Mutualiser et partager, un défi pour la génomique fonctionnelle végétale

Pierre Larmande

Numéro d'identification :

ACADÉMIE DE MONTPELLIER

UNIVERSITÉ MONTPELLIER II
— **S**CIENCES ET **T**ECHNIQUES DU **L**ANGUEDOC —

THÈSE

présentée à l'Université des Sciences et Techniques du Languedoc
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : **INFORMATIQUE**
Formation Doctorale : **Informatique**
École Doctorale : **Information, Structures, Systèmes**

Mutualiser et partager, un défi pour la génomique fonctionnelle végétale

par

Pierre LARMANDE

Soutenue le 20 décembre 2007 devant le Jury composé de :

Anne DOUCET, Professeur, Université Paris 6 ..... Rapporteur
Christine FROIDEVAUX, Professeur, Université Paris-Sud 11 ..... Rapporteur
Corine CAUVET, Professeur, Université Aix-Marseille 03 ..... Examinatrice
Thérèse LIBOUREL, Professeur, Université Montpellier II ..... Directeur de thèse
Isabelle MOUGENOT, Maître de conférences, Université Montpellier II ..... Co-encadrante
Manuel RUIZ, Chercheur, CIRAD ..... Co-encadrant

Numéro d'identification :

ACADÉMIE DE MONTPELLIER

UNIVERSITÉ MONTPELLIER II
— **S**CIENCES ET **T**ECHNIQUES DU **L**ANGUEDOC —

THÈSE

présentée à l'Université des Sciences et Techniques du Languedoc
pour obtenir le diplôme de DOCTORAT

SPÉCIALITÉ : **INFORMATIQUE**
Formation Doctorale : **Informatique**
École Doctorale : **Information, Structures, Systèmes**

Mutualiser et partager, un défi pour la génomique fonctionnelle végétale

par

Pierre LARMANDE

Soutenue le 20 décembre 2007 devant le Jury composé de :

Anne DOUCET, Professeur, Université Paris 6 ..... Rapporteur
Christine FROIDEVAUX, Professeur, Université Paris-Sud 11 ..... Rapporteur
Corine CAUVET, Professeur, Université Aix-Marseille 03 ..... Examinatrice
Thérèse LIBOUREL, Professeur, Université Montpellier II ..... Directeur de thèse
Isabelle MOUGENOT, Maître de conférences, Université Montpellier II ..... Co-encadrante
Manuel RUIZ, Chercheur, CIRAD ..... Co-encadrant

Remerciements

Je tiens tout d'abord à remercier Madame Thérèse Libourel, Professeur à l'Université Montpellier II, d'avoir accepté de diriger mes travaux de thèse durant ces cinq années.

J'exprime ma profonde et sincère reconnaissance à Isabelle Mougenot, Maître de conférences à l'Université Montpellier II, pour avoir accepté d'encadrer cette thèse. Merci Isabelle, de m'avoir soutenu et fait confiance, tes conseils et ton implication auront fortement contribué à l'aboutissement de ma thèse.

Je remercie Monsieur Manuel Ruiz, Chercheur au CIRAD, responsable de l'équipe intégration de données pour la génomique comparative (ID), de m'avoir accueilli au sein de cette dernière. Merci de m'avoir permis de concilier mon travail ingénioral et cette thèse.

Je tiens à remercier vivement Mesdames Christine Froidevaux et Anne Doucet d'avoir accepté de juger mon travail et qui par leurs remarques très constructives m'auront permis d'améliorer ce manuscrit de thèse. Je remercie également Madame Corine Cauvet d'avoir accepter de participer au jury.

Je remercie Monsieur Emmanuel Guiderdoni, Directeur de Recherche au CIRAD et responsable de l'équipe Développement Adaptatif du Riz, de m'avoir permis de développer mes travaux de recherche à partir des données expérimentales produites au sein de son équipe.

Je tiens particulièrement à remercier Gaëtan Droc, bioinformaticien au CIRAD, avec qui j'ai fait équipe depuis 5 ans sur les développements réalisés pour le riz.

Bien entendu, je remercie l'équipe ID avec qui j'ai passé de très bons moments. Je vous remercie tous également de m'avoir suppléé dans les tâches d'ingénierie système pendant cette période.

Je tiens à remercier le groupe de travail ISIBio, auquel j'ai pu participer depuis 2004. Les nombreux échanges que j'ai pu avoir m'ont énormément aidés dans mon sujet. Merci à tous les doctorants du groupe avec qui j'ai apprécié l'échange d'idées et de connaissances.

Je remercie tous mes amis Xavier, Mumu, Delphine, Jef, Fabien, Hélène, Habib, Gaëtan, Yann, Cyrille, Guillaume, Julie, Chloé, Francois.

A mes parents, merci de m'avoir soutenu dans ce parcours universitaire.

A Sophie, qui m'a supporté, soutenu, encouragé pendant toute cette période, un grand merci

A Sophie et Nina.

Sommaire

Abréviations

ADN Acide DésoxyriboNucléique

ADNc Acide DésoxyriboNucléique complémentaire

ARN Acide RiboNucléique

ARNm Acide RiboNucléique messager

BPEL4WS Business Process Execution Language For Web Services

CPL Collection Programming Language

CSS Cascading Style Sheets

EMBL European Molecular Biology Laboratory

EMBO European Molecular Biology Organisation

GAV Global As View

GLAV Global Local As View

HPG Human Genome Project

HTTP High Throughput Transfer Protocol

HTML Hypertext Markup Language

HUGO Human Genome Organisation

INSDC International Nucleotide Sequence Database Collaboration

IRS Internet Reasoning Service

JDBC Java Database Connectivity

JDO Java database Object

LAV Local As View

OASIS Organization for the Advancement of Structured Information Standards

OQL Object Query Language PNG

PCR Polymerase Chain Reaction

RT-PCR Reverse Transcription-Polymerase Chain Reaction

SOAP Simple Object Access Protocol

SRS Sequence Retrieval System

SVG Scalable Vector Graphics

SWS Services Web Semantic

TIGR The Institute for Genome Research

Abréviations

UDDI Universal Description, Discovery and Integration
XHTML The Extensible HyperText Markup Language
XML eXtensible Markup Language
W3C World Wide Web Consortium
WSMF Web Service Modeling Framework
WSDL Web Service Description Language
WSFL Web Services Flow Language
WSCI Web Services Choregraphy Interface
WSCLD Web Services Choregraphy Description Language

Introduction

La génomique fonctionnelle se définit en biologie comme un cadre dans lequel plusieurs disciplines et techniques participent à la découverte de la fonction des gènes, de leur profil d'expression, de leur régulation ainsi que de leurs interactions.

Le challenge de ces découvertes est particulièrement important pour l'ensemble de l'humanité. Pour relever celui-ci, les scientifiques doivent faire face à des difficultés diverses. La première de celle-ci réside dans la gestion des volumes importants de données générés. Pour l'instant, force est de constater que les scientifiques gèrent, au coup par coup. Toutefois, l'évolution des technologies de l'information allant vers une simplification de leur utilisation, une grande partie des données produites sont accessibles par le Web dans des systèmes souvent dédiés. La deuxième difficulté est inhérente à la recherche de corrélations pertinentes dans ces masses de données et de traitements. Le contexte de travail est donc à la croisée des domaines d'expertise biologique et informatique.

Qu'attendent les experts biologiques de la mise en place de systèmes informatisés sur leur domaine :

- qu'ils respectent leurs données, leur autonomie et l'origine en termes de propriété intellectuelle,
- qu'ils apportent une valeur ajoutée à leur travail en termes de connaissances, diffusion d'information,
- qu'ils les assistent dans leur travail de recherche (c'est sûrement un des points les plus sensibles) c'est-à-dire qu'ils facilitent et automatisent à terme les recherches, analyses, croisements d'information, etc.

Pour les scientifiques, l'intérêt principal des systèmes d'information biologiques est de pouvoir **stocker leurs propres données expérimentales** afin d'en réaliser l'**analyse** (e.g. trier, nettoyer, comparer). Le **partage et la diffusion** de l'information par le Web est aussi essentiel dans la mesure où la publication de travaux scientifiques, de nos jours, nécessite un tel accès, mais offre également un bon moyen de faire connaître son travail tout en permettant de créer des collaborations scientifiques. En effet, il est difficile pour les scientifiques d'exploiter la totalité des données produites. Pour donner un exemple, l'étude de la fonction d'un gène responsable d'une mutation chez le riz (*Oryza sativa*) peut prendre jusqu'à 3 à 4 ans. Certaines équipes constituent des collections de mutants dans l'objectif d'interrompre le fonctionnement de l'ensemble des gènes ce qui représente au minimum 34 000 mutants chez le riz. Ces équipes choisissent de ne travailler que sur une partie de ces mutants, généralement dans les domaines où elles excellent, et partagent le reste des ressources avec d'autres équipes. Enfin, ces systèmes informatiques permettent de **protéger l'accès** aux données. Tout d'abord, dans un cadre légal sur la protection des données et des programmes (par exemple les données diffusées ne peuvent être utilisées qu'avec le consentement du fournisseur). Ensuite, parce qu'une source de données peut être accessible avec **différents niveaux de confidentialité**, autorisant ainsi une diffusion rapide des données sans pénaliser les scientifiques qui souhaitent valoriser une partie de leur travail.

La détermination de la fonction des gènes requiert souvent l'utilisation séquentielle de plusieurs sources de données. Prenons l'exemple du transfert des connaissances entre deux espèces végétales : le riz (*Oryza sativa*) et l'arabette des dames (*Arabidopsis thaliana*). Nous allons, ainsi, pouvoir transférer des connaissances acquises sur l'espèce la plus étudiée, en l'occurrence *Arabidopsis thaliana* vers l'espèce la moins étudiée, ici *Oryza sativa*. Plus précisément, le gène ERECTA (Gene id : At2g26330 dans la source TAIR) est connu chez *Arabidopsis thaliana* pour être impliqué, entre autres, dans les mécanismes de régulation de l'efficience de la transpiration chez Arabidopsis [MGF05]. résistance à un pathogène *Ralstonia solanacearum*. De nombreuses descriptions phénotypiques sont disponibles dans la base TAIR, avec des liens vers des publications (source PUBMED). En ce qui concerne *Oryza sativa*, un gène potentiellement orthologue du gène ERECTA (entrée Os06g10230 dans la source OryGenesDB) a été identifié avec pour toute information disponible, sa séquence nucléique annotée. Il va s'agir alors de réutiliser les connaissances acquises en génomique fonctionnelle autour d'ERECTA (caractérisation de la variabilité d'expression, localisation spatiale et temporelle de l'expression du gène, gènes co-exprimés, etc), en supposant que ces connaissances sont valides dans le contexte d'*Oryza sativa*.

La recherche d'information nécessaire pour valider de telles hypothèses nécessite une navigation Web à travers de nombreuses sources. Dans ce contexte, la conception de systèmes intégrant plusieurs sources de données permet de réduire les temps de recherche navigationnelle dans les ressources du Web. De plus, rassembler des données complémentaires, permet d'avoir une vision globale du domaine étudié mais aussi d'effectuer des traitements sur des données regroupées en utilisant un seul langage de manipulation. Dans le même sens, rassembler des données chevauchantes (dont le domaine d'étude est proche, voir l'exemple du gène ERECTA), permet de transférer ou confirmer des résultats expérimentaux (par exemple, si les résultats sont similaires chez l'arabette qui est mieux étudiée ou si d'autres sources spécifiques du riz trouvent des résultats similaires). Enfin, l'intégration permet de générer de nouvelles connaissances grâce aux déductions qui peuvent être établies à partir de l'ensemble des don-

nées mises à disposition (par exemple étude et la simulation de systèmes biologiques).

Les propositions actuelles, les plus à même de répondre aux diverses fonctionnalités présentées ci-dessus, relèvent de la thématique du domaine systèmes d'information, bases de données et plus précisément des approches dites intégration et médiation de données. Notre travail s'inscrit donc dans ce contexte. L'hétérogénéité et la complexité des données constituent un des écueils (bien identifié) et celui-ci est de plus augmenté de part l'évolutivité inhérente au domaine de la génomique fonctionnelle.

L'objectif de cette thèse est de contribuer à l'automatisation de l'accès uniforme et transparent aux informations issues de plusieurs sources de données de génomique fonctionnelle tout d'abord en partant de projets et expériences en cours pour aboutir sur un cadre méthodologique dédié au domaine.

Ce document est organisé en trois parties.

Dans la première, nous étudions le contexte général du travail. Cette partie comprend trois chapitres. A partir d'une présentation rapide (chapitre1) de ce que représente la génomique fonctionnelle et les recherches menées dans ce domaine, nous arrivons au travers de divers exemples, à montrer l'importance de la mise en œuvre de services informatique pour pallier les divers besoins en termes de mutualisation, partage et intégration. Dans le chapitre 2, nous dressons un état des lieux sur le partage des données biologiques et les efforts effectués par la communauté dans la standardisation de la représentation des données, tout en relevant les problèmes encore d'actualité. Enfin, le chapitre 3 est consacré à un rappel des caractéristiques et principes généraux liés à l'intégration de données puis à un état de l'art relatif aux systèmes d'intégration existants en bioinformatique.

La deuxième partie est consacrée aux diverses approches et expériences que nous avons menées dans le domaine de l'intégration de données de génomique végétale. Cette partie comprend trois chapitres. Dans le chapitre 4, nous décrivons la mise en place de sources de données candidates à l'intégration au cours de divers projets en génomique fonctionnelle : Oryza Tag Line, une base de données phénotypique sur la collection de mutants d'insertion chez le riz et OryGenesDB, une base de données intégrative pour les données génomiques du riz. Les deux autres chapitres relatent les deux approches adoptées pour l'intégration afin d'en évaluer ultérieurement les avantages et les inconvénients. Au chapitre 5, nous présentons les résultats de l'adaptation d'un système médiateur sur les sources présentées dans le chapitre 4. Nous décrivons tout d'abord les caractéristiques du système, puis nous présentons les résultats de l'intégration avec les solutions apportées pour résoudre les problèmes d'hétérogénéité. Dans le chapitre 6, nous décrivons le développement d'un environnement utilisateur personnalisé intégrant les sources présentées dans le chapitre 4 à travers des enchaînements de services Web. Nous aborderons les principes de fonctionnement des services Web, puis nous décrivons les caractéristiques de BioMOBY-S, un projet d'annuaire de services Web bioinformatiques que nous avons utilisé. Enfin, nous détaillons les développements réalisés pour la conception des services et leur chorégraphie.

Dans la troisième et dernière partie, nous réaliserons une synthèse des approches que nous avons réalisées en mettant en évidence nos contributions. Nous présentons une méthodologie d'intégration adaptée aux besoins de la génomique fonctionnelle végétale et les perspectives que celle-ci ouvre.

Première partie

Contexte et état de l’art
Mutualiser et partager, les objectifs de base

Chapitre 1

Du Gène à la fonction

Sommaire

1.1 Introduction

La naissance de la génétique moderne se situe en 1865, date des publications de Mendel. A l'époque, ses travaux passent inaperçus. Il définit pourtant les règles qui régissent la transmission des caractères héréditaires et définit sans le nommer les propriétés du gène. Les travaux de Mendel vont influencer de nombreux scientifiques au début du 20ieme siècle, notamment Morgan. Ses travaux sur la drosophile conduisent aux théories sur l'hérédité [Mor10, MSMB15], et démontrent que chaque parent contribue pour moitié au patrimoine génétique de la descendance. Dans ses expériences, Morgan montrera que l'arrangement des gènes sur les chromosomes est linéaire. En 1912, l'hypothèse selon laquelle les chromosomes sont le support de l'hérédité est complètement acceptée. Les généticiens s'attachent dès lors à découvrir les caractères génétiquement transmissibles et à les cartographier. Les processus qui conduisent de la mutation du gène au phénotype restent encore une énigme. C'est Garrod qui en travaillant sur des mutants liés au métabolisme de la phénylalanine, émet le premier, l'hypothèse de l'existence d'une relation gène-enzyme [Gar08]. Ce sont les expériences de Beadle et Tatum avec le champignon *Neurospora crassa*, qui confirment cette hypothèse en 1941 [BT41]. Peu de temps après, Avery met en évidence le fait que certains caractères phénotypiques du pneumocoque peuvent être transmis par son ADN, démontrant que l'ADN est effectivement le support moléculaire de la transmission de l'hérédité [AMM44]. D'autres découvertes prennent date dans l'histoire des sciences. En 1953, Watson et Crick découvrent la structure en double hélice de l'ADN (Acide Désoxyribo Nucléique) et suggèrent un mécanisme de réplication de l'ADN [WC53](figure 1.1c) . En 1961, Nirenberg et Mattaei découvrent le code génétique universel [MMJN62, MJMN62], tandis que Jacob, Monod et Lwoff élucident le rôle des ARN messagers et le mécanisme de régulation de l'opéron lactose [JM61].

Comme le montre la figure 1.1, le chromosome, et, par extension le génome, peut être étudié selon plusieurs angles de vue. La molécule d'ADN a tout d'abord été étudiée avec une vision macromoléculaire et par ses propriétés biochimiques. Par exemple, le comptage des chromosomes par microscopie ou sa coloration. Mais l'élément qui fut déterminant dans l'étude du génome fut l'utilisation des enzymes de restrictions au début des années 70. Ils permettent de manipuler la molécule d'ADN en coupant, collant, dupliquant des morceaux. Cela a permis d'établir entre autre les cartes de restrictions, première étape de l'établissement de cartes physiques (figure 1.1d). Les cartes physiques ont pour objectif de reconstituer l'ensemble du génome au niveau de détail le plus fin : la base nucléotidique. Les cartes génétiques (figure 1.1a) servent à encadrer des gènes ou des fonctions biologiques majeures quantifiables par le biais de marqueurs séparés par des distances génétiques basées sur des taux de recombinaison. L'établissement de telles cartes permet d'avoir une vision globale de la répartition des gènes.

Deux découvertes techniques, le séquençage et la PCR, ont permis d'entrevoir le décryptage des génomes comme quelque chose de réalisable. Dès la fin des années 70, deux méthodes de

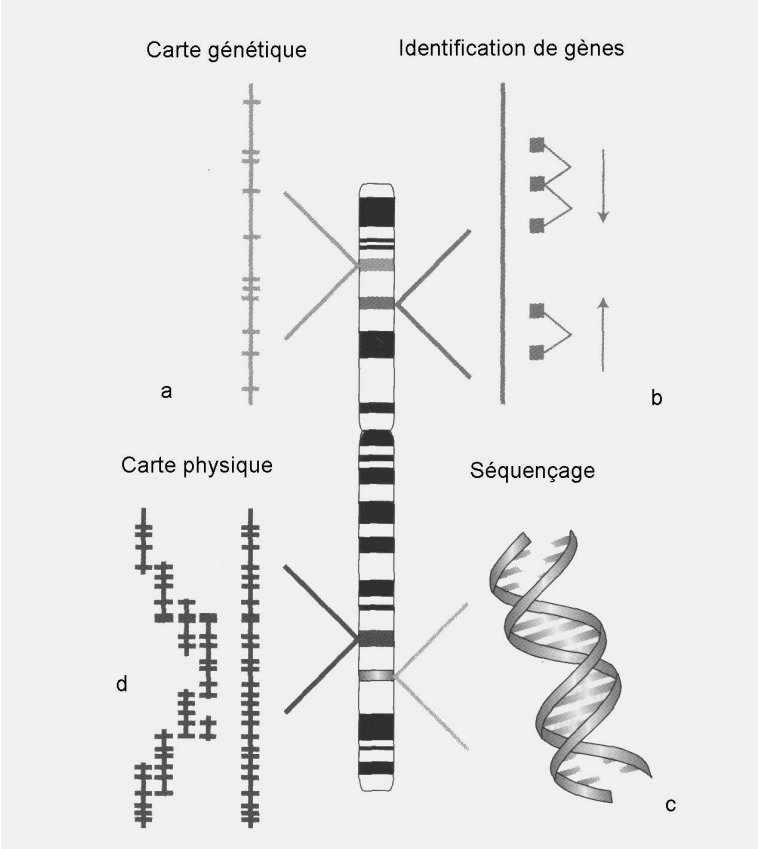


FIG. 1.1 – Différentes représentations d'un chromosome (adapté de Collins et al, 1993 [CG93]).

séquençage, enzymatique [SNC77] et chimique [MG77] permettaient d'identifier chaque base d'une courte séquence d'ADN de quelques centaines de bases. Moins toxique, la première technique fut utilisée par la suite. En 1984 apparaît la PCR (Polymerase Chain Reaction)([MFS+86], une technique qui va révolutionner la biologie moléculaire . Cette technique d'amplification de l'ADN a été rendue possible grâce à la découverte d'une enzyme polymérase fonctionnant à des températures élevées. Des techniques dérivées de la PCR ont permis de mettre au point et d'utiliser une très grande panoplie de marqueurs moléculaires permettant l'enrichissement des cartes génétiques. Dès lors, il devenait théoriquement envisageable de séquencer un génome en le fragmentant en sous partie ordonnées (i.e. carte physique 1.1d) qui étaient séquencées. Il faudra attendre 1987, et la commercialisation du premier séquenceur automatique analysant 96 échantillons simultanément pour que l'idée soit potentiellement réalisable.

Le projet de séquençage du génome humain débute en 1990 avec la création du HGP[1] (Human Genome Project). Le projet est soutenu et coordonné par le consortium HUGO[2] (Human Genome Organisation) [McK89]. Les premiers génomes séquencés arrivent par la suite. En 1995, l'équipe de Graig Venter au TIGR[3] (The Institute for Genome Research) publie le premier génome complet d'une bactérie *Haemophilus influenzae* (1,8 Méga Bases MB), réalisé par une technique dite de shotgun, de séquençage aléatoire et reconstitution *in silico* du génome [FAW+95]. Le génome de la levure (16 Mb) *Saccharomyces cerevisiae* séquencé par un consortium international sera terminé en 1996 [Gof96] alors que le premier génome d'eucaryote pluricellulaire, le ver nématode *Caenorhabditis elegans* [eSC98] sera publié un an plus tard. Les génomes de l'insecte *Drosophila melanogaster* [ACH+00] et la plante *Arabidopsis thaliana* (130Mb) [Ini00], seront publiés la même année en 2000. C'est en février 2001 que la séquence de 95% du génome humain est publiée conjointement par le HPG [Con] et la société Celera Genomics[4] [VAM+01]. 2004 sera l'année de la publication du génome du riz *Oryza sativa* variété Niponbarre (430 Mb) [Int05a].

1.2 La génomique végétale et le riz

1.2.1 Les caractéristiques agronomiques du riz

Le riz est la deuxième céréale après le maïs en termes de surface cultivée (153 Mha en 2004) et de quantité produite (608 Mt en 2004), avec un rendement moyen de 4,0 t/ha qui masque de très importantes disparités [Sta05]. C'est, en revanche, la première céréale pour l'alimentation humaine avec des consommations annuelles très importantes dépassant dans certains pays en voie de développement les 100 kg/habitant. L'Asie domine l'économie du riz avec 90 % des surfaces et de la production qui y sont concentrées, l'Amérique Latine et l'Afrique se partageant l'essentiel des 10 % restants. Le riz est avant tout une production d'autoconsommation, les grands pays producteurs (Inde, Chine, Indonésie, Bangladesh, Thaïlande, Vietnam) étant également les principaux consommateurs [Cou07].

[1] http ://www.ornl.gov/sci/techresources/Human_Genome/home.shtml
[2] http ://www.hugo-international.org/
[3] http ://www.tigr.org/
[4] http ://www.celera.com

Le riz est cultivé dans des milieux très variés couvrant une large gamme d'altitudes et de latitudes. Cette plante, d'origine aquatique, et donc assez exigeante en eau par rapport à d'autres céréales, est surtout caractérisée par une grande plasticité vis-à-vis de ses conditions d'alimentation hydrique. C'est sur ce point que se fondent la plupart des classifications des types de rizicultures [Cou88] :

- Riziculture irriguée, endiguée, avec parfaite maîtrise de l'eau qui occupe 53 % des surfaces.
- culture inondée, endiguée, sans maîtrise de l'eau. Ce type de riziculture représente 25 % des surfaces.
- Riziculture pluviale dont l'alimentation hydrique dépend uniquement de la pluviométrie ou de la présence d'une nappe éventuelle. Ce type de riziculture représente 13 % des surfaces en Asie mais respectivement 60 et 75 en Afrique et en Amérique Latine.
- Riziculture flottante, qui suit la crue des grands fleuves, occupant 9 % des surfaces.

Les rendements augmentent avec le degré de maîtrise de l'eau. En revanche, l'aménagement des rizières pour permettre l'irrigation, augmente les coûts de productions. La riziculture irriguée permet une intensification de la culture (double voire triple culture annuelle) et une diminution appréciable des aléas de culture garantissant des rendements élevés (6 t/ha en saison des pluies et jusqu'à 10 t/ha en saison sèche). La culture pluviale, en revanche, ne demande aucun aménagement particulier mais comporte plus de risques, notamment en cas de sécheresse. La production n'est répartie que sur un seul cycle de culture et les rendements sont plus faibles et plus variables (entre 1,5 t/ha et 4,5 t/ha).

D'un point de vue botanique, le riz est une Monocotylédone de la famille des Poacées. Deux espèces sont cultivées, *Oryza sativa* (génome A, 2n=24) à distribution mondiale et *Oryza glaberrima* (génome A, 2n=24), cantonnée à l'Afrique de l'Ouest [Int05b]. *Oryza sativa* est une céréale autogame (moins de 1% d'allogamie d'après [Cha64]). Les structures génétiques traditionnellement cultivées sont donc essentiellement des lignées pures. La diversité génétique du riz est considérable avec plus de 150.000 variétés cultivées dans le monde et 107.000 accessions environ dans la banque de gènes de l'IRRI. C'est une espèce fortement bipolaire avec 2 groupes d'origines géographiques différentes, les indicas et les japonicas, clairement distingués sur la base de caractéristiques agromorphologiques, de comportement en croisement, et de marqueurs biochimiques et moléculaires. La recombinaison entre les deux groupes n'est ni facile ni fréquente.

Le potentiel génétique a fait un bond exceptionnel à la fin des années 50 avec la découverte d'un mutant naturel semi-nain, Dee-geo-woo-gen, qui a été depuis très largement utilisé comme donneur de semi-nanisme. L'amélioration génétique du riz aboutit à la création de variétés à cycle court, ayant de bons rendements, une bonne qualité de grains ainsi que des qualités de résistances multiples à des insectes ou à des agents pathogènes. Une des variétés les plus utilisée en Asie, IR64, fut vulgarisée en 1985.

Le développement des techniques d'analyse moléculaire à la fin des années 80 s'est traduit par un changement d'échelle dans les analyses génétiques. La première carte génétique du riz, établie par McCouch et al. [MKZ+88] à partir d'une population F2 indica x japonica, a été suivie par beaucoup d'autres. Les cartes génétiques ont permis de cartographier de nombreux gènes majeurs de résistance à des maladies ou des insectes. Elles ont, en outre, été utilisées

pour déterminer la localisation dans le génome des loci contrôlant des caractères quantitatifs complexes (QTLs) de grand intérêt pour les sélectionneurs : résistance partielle à la pyriculariose, croissance racinaire, composantes du rendement, notamment. Le premier article sur la détection de QTLs de résistance à la pyriculariose chez le riz a été publié par Wang et al [WMB+94]. Depuis, des QTLs ont été détectés pour de nombreux caractères (Xu, 2002, pour une revue [Xu02]). Ces progrès dans le marquage de gènes utiles permettent désormais d'envisager la mise en place de sélection assistée par marqueurs. L'existence de ces cartes génétiques saturées puis le développement de banques BAC ont rendu possible le clonage positionnel des gènes majeurs intéressants comme Xa21, qui détermine la résistance au flétrissement bactérien [SWC+95], sd1, le gène de semi-nanisme [SAUT+02], ou récemment le gène codant pour le composé majeur de l'arôme du riz [BFH+05]. Les premiers QTLs de riz ont été clonés par Yano et al [YKA+00] en partant d'un jeu de lignées de substitution. Ces QTLs contrôlaient la durée de cycle, un caractère facile à phénotyper. Mais la gamme des QTL clonés s'est rapidement élargie à des caractères plus complexes, comme le nombre de grains par panicule [ASL+05]. Les QTLs clonés correspondent parfois à des mutations connues (cas du nombre de grains par panicule) et ces résultats semblent suggérer qu'une meilleure exploitation des ressources génétiques caractérisées pourrait être profitable.

1.2.2 Le riz, une espèce modèle pour les Poacées

En plus des aspects économiques et agronomiques le riz a été retenu comme espèce modèle sur la base de plusieurs critères :

Les caractéristiques de son génome C'est la céréale qui a le plus petit génome. Constitué de 12 paires de chromosomes, son génome contient peu de séquences répétées par rapport aux autres céréales (voir figure 1.1).

Les relations entre le génome du riz et celui des autres céréales En 1995, un modèle général de l'organisation des différents blocs chromosomiques de céréales a été présenté [Moo95], selon lequel chaque génome est représenté par des cercles concentriques permettant d'établir la colinéarité entre segments de chromosomes des différentes céréales (figure 1.2). La conservation de l'ordre des gènes sur un chromosome d'une espèce à l'autre est appelé synténie. Comme on peut le voir sur cette figure, certains chromosomes de blé correspondent à des mosaïques de plusieurs chromosomes de riz (ch 5 et 7 de blé). Ce modèle en cercle concentrique permet de reporter des informations d'une espèce à une autre et de faire des prédictions. Par exemple, certains gènes et QTLs sont communs à différentes céréales, et leurs positions sur les cartes apparaissent à l'intersection d'un rayon avec les différents cercles. En théorie, si le principe de colinéarité est conservé entre blocs synténiques, nous pouvons retrouver un gène homologue à la même position chez les différentes céréales. C'est le cas pour le gène de nanisme *sd1*. Ce type d'observation est intéressant pour les espèces dont le génome n'est pas encore séquencé (i.e. blé ou maïs) car le fait d'avoir un gène homologue ou de nouveaux marqueurs facilite l'isolement du gène d'intérêt dans l'espèce cible (i.e. stratégie du clonage positionnel, section 1.4.2).

La disponibilité d'outils pour l'analyse génétique Tous les outils créés visent à l'identification du rôle de chaque gène. Il y a tout d'abord l'annotation fonctionnelle, exécutée par des programmes bioinformatiques qui, de plus en plus, s'appuient sur des ressources biologiques (e.g. ADNc pleine longueur). Le riz ayant la particularité d'être facilement transformable par *Agrobactérium* (voir section 1.3.1), de nombreuses collections de mutants

ont été développées de par le monde pour essayer d'identifier la fonction de tous les gènes. Couplées à des analyses phénotypiques, cette stratégie permet d'identifier des plantes mutantes pour un gène précis.

Arabidopsis thaliana	130
Riz	430
Medicago truncatula	650
Brassica oleracea (chou)	700
Sorgho	800
Tomate	1000
Colza	1200
Maïs	2500
Pois	5000
Orge	5200
Blé	16 000
Tulipe	30 000

TAB. 1.1 – Taille des génomes de différentes espèces végétales en millions de paires de bases (pb)

1.2.3 Le séquençage du génome du riz

Le séquençage du riz débute en 1999 coordonné par l'IRGSP (International Rice Genome Sequencing Project)[5]. L'IRGSP, un consortium public dont l'objectif est de séquencer de manière exhaustive, clone par clone, le génome de la variété Nipponbare d'*Oryza sativa* L. *ssp japonica* obtiens une séquence complète en 2004 [Int05a]. Parallèlement, un groupe de Pékin en Chine séquence une variété de l'autre sous-espèce de riz, la sous-espèce *indica* [YHW+02, JJW+05], avec une technique différente, celle du shot gun. Durant cette période, deux sociétés Monsanto et Syngenta réalisent un séquençage complet de type shot gun de la variété Nipponbare [GRL+02]. Parmi les séquençages réalisés, celui effectué par l'IRGSP se distingue par le fait qu'il reste très peu de "trous" et qu'il ne s'agit pas d'un séquençage *shot gun* puisque toutes les séquences de BACs sont ancrées sur la carte génétique de référence du riz.

Le nombre de gènes est estimé à 34.000 (pour 28.000 ADNc pleine longueur connus) avec très peu de différence en contenu en gènes entre les génomes indica et japonica mais des différences intergéniques massives [JJW+05].
Ces séquences ont permis d'accéder directement aux gènes. Des programmes de génomique fonctionnelle, tel celui du Cirad conduit par E. Guiderdoni, se sont mis en place afin de déterminer leur fonction, inconnue dans plus de 60 % des cas [HGA+04]. L'accès à l'ensemble des gènes cependant n'est qu'un préalable. Il faut ensuite déterminer les gènes pertinents dans le contexte agronomique visé par le biais d'études d'expression et, parmi ceux-ci, identifier ceux qui sont responsables de la variation phénotypique observée dans l'espèce. Il faut en analyser le polymorphisme et identifier les allèles favorables qui pourront être réunis par les sélectionneurs dans des variétés élites. Ces travaux sur la diversité fonctionnelle ont commencé chez le riz et devraient voir d'importants développements dans les années à venir.

[5] http ://rgp.dna.affrc.go.jp/IRGSP/

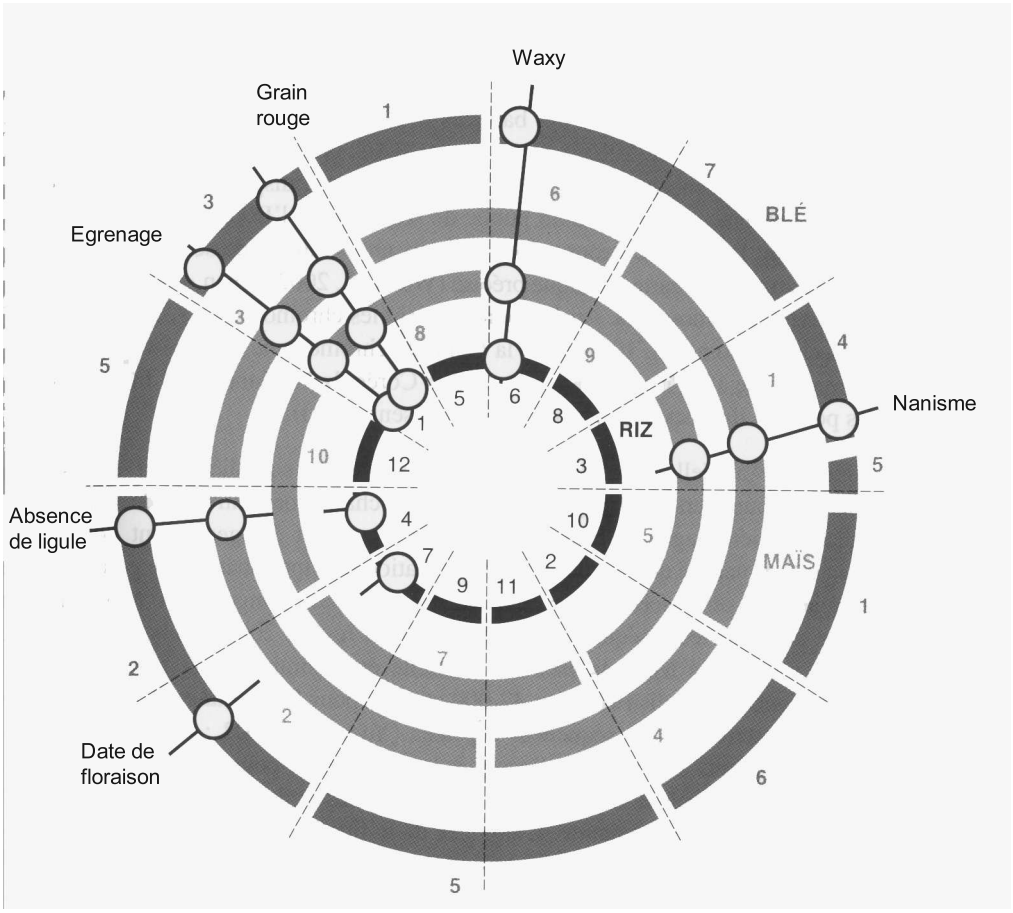


FIG. 1.2 – Représentation schématique des génomes de riz, blé et maïs, disposés selon le modèle des cercles concentriques. Dans le cas du génome du blé, seul l'un des trois génomes est représenté, ceux-ci étant complètement colinéaires. Les rayons représentés en pointillés correspondent aux limites des chromosomes du riz. Les positions de quelques gènes d'intérêt agronomique sont alignées sur les rayons du modèle (d'après Devos et Gale, 1997). (Adapté de Delseny et al., [MGB04])

1.3 La génomique fonctionnelle

La génomique fonctionnelle est l'étude de la fonction des gènes. C'est le fait d'assigner un rôle biologique aux gènes. Ces travaux sont complémentaires à l'annotation d'un génome. L'annotation *ab initio* ne permet pas la plupart du temps de découvrir la fonction d'un gène. L'étude de mutants joue dans ce cas un rôle important. Depuis les premiers travaux de Mendel, l'étude des variants isolés dans les populations naturelles ou induits par mutagénèse, ont permis de localiser et d'identifier des gènes pris individuellement, puis d'en déterminer la fonction biologique. Mais le gène muté était, jusqu'il y a peu, difficile à localiser. Il faudra attendre le début des années 90 pour que les premières expériences de mutagénèse insertionnelle chez les plantes puissent laisser entrevoir un avenir autre que théorique pour la génomique fonctionnelle.

1.3.1 La mutagénèse

Il existe 3 types de mutagénèse : la mutagénèse chimique (EMS), la mutagénèse physique (rayon X, neutrons, etc) et la mutagénèse insertionnelle. Alors que les deux premières provoquent des modifications de l'ADN par transformation ou délétion de bases, la mutagénèse insertionnelle introduit un élément d'ADN étranger dans le génome hôte. Cette dernière technique s'est beaucoup développée parce que l'élément étranger sert d'étiquette afin de localiser directement la mutation (figure 1.3). Dès lors, est venu l'idée de produire des collections de mutants, c'est à dire des ensembles de plantes indépendantes portant chacune une (de préférence) ou plusieurs insertions de l'élément mutagène. Plus la population de mutants est importante plus on augmente la probabilité d'inactiver chaque gène du génome. Cette saturation du génome dépend de la taille moyenne d'un gène, de la taille et de l'organisation du génome de la plante utilisée et du nombre d'insertions dans la population [BH98]. Pour que la saturation soit efficace il est important que les insertions aient lieu au hasard et que leur nombre par lignée reste faible. Les populations peuvent être utilisées en crible direct - isoler un gène à partir d'un phénotype observé - et/ou en crible inverse - partir d'un gène connu inactivé pour en observer le phénotype. La génétique inverse à grande échelle permet d'isoler un grand nombre de lignées présentant des gènes affectés et d'étudier leur fonction biologique.

1.3.2 Les différents types de mutagénèse insertionnelle

1.3.2.1 L'ADN-T

L'ADN-T est une petite séquence d'ADN présente dans le plasmide Ti (tumor inducing) de la bactérie *Agrobacterium tumefaciens* (figure 1.3). Dans la nature, cette bactérie du sol phytopathogène est capable de transférer et d'insérer ce fragment dans le génome d'une plante, déclenchant ainsi une tumeur. Cette fonctionnalité a été utilisée en laboratoire afin d'introduire de l'ADN étranger dans le génome des plantes (figure 1.3). Les seuls éléments nécessaires aux transferts du fragment sont ses extrémités gauche et droite de 25-pb, le reste de la séquence est souvent remplacé par des éléments de sélection génétique. Le T-DNA peut être également équipé de gènes rapporteurs qui, lorsqu'ils s'expriment, permettent de visualiser sa localisation dans la plante (e.g. fluorescence pour le gène de la GFP, coloration bleu pour le gène GUS). Lorsque le fragment "étiquette"(ou "Tag") est inséré au hasard dans un gène, il provoque une mutation stable dans la descendance avec, éventuellement, des phénotypes observables. Les premières expériences concluantes chez la sous-espèce japonica de riz ont été réalisées [CCH$^+$93].

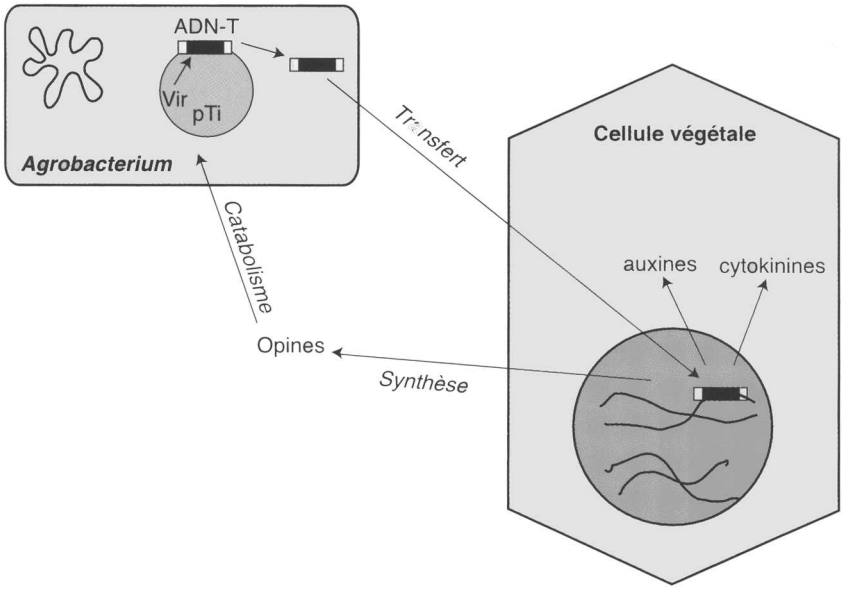


FIG. 1.3 – Schéma représentant le transfert d'ADN par *Agrobactérium*

Il faudra cependant attendre quelques années pour que ces protocoles soient appliqués dans des conditions de haut débit [SLR+03, TAI04]).

1.3.2.2 Les transposons

Également appelés éléments transposables, ce sont des séquences d'ADN endogènes capables de se déplacer de manière autonome dans le génome par un mécanisme que l'on nomme transposition. Présents chez presque tous les organismes vivants, ils sont un des constituants important des génomes eucaryotes. Par exemple ils représentent près de 70% du génome du maïs et sont en partie responsables de l'accroissement de la taille du génome des plantes. Le riz possède un transposon endogène T*os 17* qui se déplace dans le génome au cours de la transformation des cals par *Agrobactérium*. Une des méthodes de génétique inverse consiste à localiser les séquences T*os 17* qui ont bougé et perturbé l'expression de gènes. D'autres méthodes empruntées au maïs consistent à inclure dans le T-DNA des systèmes transposons tels que *Ac/Ds (Activator/Dissociation)* ou *En/Spm (Enhencer/Suppressor-Mutator)*. Ces systèmes ont la particularité en plus de l'insertion au hasard dans le génome procurée par le T-DNA d'être à nouveau excisé du gène interrompu. C'est l'étape de remobilisation du transposon. Par exemple, en présence d'une transposase apportée par croisement (enzyme provoquant la transposition) il est possible de remobiliser le système *Ac/Ds (Activator/Dissociation)*, ce qui aura pour effet de générer de nouvelles insertions dans la zone proche de l'insertion de départ.

1.3.3 Utilisation des collections d'insertion pour la génétique directe et inverse

Aujourd'hui l'enjeu biologique est différent de celui d'hier. Après l'obtention des séquences génomiques complètes ainsi que la localisation des gènes sur les chromosomes, il reste à définir leurs rôles dans l'organisme. Une séquence n'a de valeur que si elle est attachée à une infor-

mation. Il est possible d'attribuer une fonction putative à un gène par similitude avec des gènes connus mais si la comparaison de séquences avec celles des banques permet d'assigner le plus souvent la fonction biochimique, elle ne donne pas le rôle spécifique du gène dans l'organisme. Une fonction putative a pu être assignée à 70% des 26 000 gènes d'*Arabidopsis* mais celle de 30% d'entre eux reste encore totalement inconnue soit parce qu'ils n'ont aucune similitude avec des séquences connues, soit parce qu'ils sont similaires à des séquences inconnues (AGI, 2000). En complément des analyses *in silico* doivent donc être menées des études expérimentales. Par génétique directe, l'observation d'un phénotype mutant est le moyen le plus simple pour accéder à la fonction du gène muté. La mise en place de stratégie de crible direct sur des collections entières permet d'obtenir de nombreuses informations exploitables par la suite. Ces stratégies sont couplées également à des stratégies de génétique inverse, dans lesquelles toutes les insertions de la collection sont caractérisées et positionnées sur le génome [SGL+04]. Dans ce cadre, la mise en place de systèmes d'information permet d'effectuer des recherches dans ces collections par génétique inverse [DRL+06] ou directe [LGL+07].

1.4 Le besoin d'accès à des multiples sources

Comme les sections précédentes le laissent deviner, la recherche en génomique fonctionnelle nécessite un accès à de multiples ressources (i.e. génomes, cDNA, protéines, collections de mutants, cartes génétiques, etc). Parallèlement au séquençage et à l'annotation du génome du riz, de nombreux laboratoires à travers le monde tentent de caractériser, par des méthodes différentes et complémentaires, la fonction des 34,000 gènes prédits. La plupart des ressources créées (bases de données, outils d'analyses, ressources génétiques) sont issues de la recherche publique et donc disponibles librement. Ce libre accès donne bon nombre d'ouvertures, en terme de recherches à mener, pour les biologistes ; mais se révèle également une difficulté dans la nécessité d'accéder, de manière séquentielle et ordonnée, à différentes sources de données afin d'enrichir l'information sur la fonction des gènes du riz. 1.4,1.5,1.6 sont des inventaires partiels des ressources mises à la disposition des chercheurs en génomique fonctionnelle du riz.

Le tableau 1.4 est une synthèse non exhaustive des ressources dédiées à l'étude du génome du riz. Il est à noter que les sources de données généralistes, comme par exemple GenBank ou SwissProt, se révèlent également d'intérêt pour l'étude du génome du riz. Les sources de données spécialisées apportent, cependant, une valeur ajoutée en terme d'une part de richesse et de qualité des expertises associées à l'information expérimentale, et d'autre part, en terme de rapidité de mise à disposition des informations. Pour exemple, lorsque le génome du riz japonica était en cours de séquençage, les sources de données génomiques IRGSP, TIGR et RiceGAAS étaient préférées à GenBank, par les biologistes et bioinformaticiens, car les mises à jour y étaient plus fréquentes. Pour autre exemple, certains points de vue ou expertises sur les données sont difficilement pris en charge par les sources généralistes. Des sources de données comme OryzaSNP, MOsDB, Gramene, KOME, Rice proteome ou encore OMAP vont se consacrer à la représentation de cette diversité de points de vue et d'expertises pour ce qui concerne le riz. De nombreuses collections de mutants (tableau 1.5) sont également disponibles, mais, il n'existe pas encore de portail commun pour le génome du riz, à l'image de TAIR pour d'*Arabidopsis* , qui unifierait l'accès aux différents sources de données. A la décharge de la communauté travaillant sur le riz, le projet de sequençage du riz est plus récent ; les conditions d'accès aux sites sont encore souvent réglementées par des authentifications et les demandes de matériels biologiques sont également, fréquemment, soumises à des agréments. Dans le do-

Database	Description	URL
RGP	Japanese Rice Genome Research Program	http://rgp.dna.affrc.go.jp/E/
IRGSP	International Rice Genome Sequencing Project	http://rgp.dna.affrc.go.jp/E/IRGSP/index.html
RAP-DB	Rice Annotation Project database (IRGSP-assembled pseudomolecules)	http://rapdb.lab.nig.ac.jp/
INE	Integrated Rice Genome Explorer	http://rgp.dna.affrc.go.jp/E/giot/INE.html
RiceGAAS	Rice Genome Automated Annotation System	http://ricegaas.dna.affrc.go.jp/
RAD	Rice Annotation Database	http://golgi.gs.dna.affrc.go.jp/SY-1102/rad/
TIGR Rice	The Institute of Genomic Research (TIGR) rice genome annotation database (TIGR-assembled pseudomolecules)	http://www.tigr.org/tdb/e2k1/osa1/
Oryzabase	Oryza genetics database	http://www.shigen.nig.ac.jp/rice/oryzabase/top/top.jsp
IRIS	International Rice Information System	http://www.iris.irri.org/
IRFGC	International Rice Functional Genomics Consortium	http://www.iris.irri.org:8080/IRFGC/
BGI RISe	Beijing Genomics Institute Rice Genome Database	http://rise.genomics.org.cn/rice/index2.jsp
OryzaSNP	Oryza Single Nucleotide Polymorphism Consortium	http://www.oryzasnp.org/
OMAP	Comparative Genome Maps of *Oryza* wild relatives	http://mips.gsf.de/proj/plant/jsf/rice/index.jsp
Rice Proteome	Rice Proteome Database	http://gene64.dna.affrc.go.jp/RPD/main_en.html
KOME	Rice full-length cDNA sequence database	
RicePIPELINE	Unification tool for NIAS rice databases	http://cdna01.dna.affrc.go.jp/PIPE/
Gramene	Comparative grass genomics anchored on rice	http://www.gramene.org/
MOsDB	Munich Information Center for Protein Sequence (MIPS) (*Oryza sativa*) Database	http://mips.gsf.de/proj/plant/jsf/rice/index.jsp

FIG. 1.4 – Bases de données génomiques spécifiques du riz (adapté de Antonio et al. [ABY+07])

Database	Description	URL
Tos17	*Tos17* insertion mutant database	http://tos.nias.affrc.go.jp/
OryGenesDB	Rice flanking sequence tags	http://orygenesdb.cirad.fr
Oryza Tag Line	Rice phenotypic and reporter gene expression database	http://urgi.infobiogen.fr/OryzaTagLine/
Rice Mutant Database	Rice mutant database for T-DNA insertion lines	http://rmd.ncpgr.cn/
Rice *Ds* Tagging Lines	Rice *Ac*/*Ds* mediated gene tagging lines	http://genebank.rda.go.kr/dstag/
TRIM	Taiwan Rice Insertional Mutants	http://trim.sinica.edu.tw/index.php
SHIP	Shanghai T-DNA Insertion Population	http://ship.plantsignal.cn/index.do
RISD	Rice T-DNA Insertion Sequence	http://an6.postech.ac.kr/pfg/index.php
Rice FST Database	Rice insertion lines containing *Ds* gene trap, *Ds* element or *dSpm*	http://sundarlab.ucdavis.edu/rice/blast/blast.html
CSIRO Resources	*Ds*/T-DNA launch pads and *Ds* insertion lines and FSTs	http://www.pi.csiro.au/fgrttpub/knowngene.htm
RiceGE	Rice Functional Genomics Browser	http://signal.salk.edu/cgi-bin/RiceGE

FIG. 1.5 – Bases de données de mutants (adapté de Antonio et al. [ABY+07])

maine du transcriptome , les données sont disponibles soit au sein des banques généralistes (e.g. GenBank, dbEST), soit au sein de sources de données spécialisées (tableau 1.6) qui sont en général associées à une technologie expérimentale spécifique (puces Affymetrix, puces CGH, technique SAGE, RT-PCR, ...). En conséquence, ces sources proposent souvent, et de manière complémentaire, des outils de visualisation et/ou de fouille de données adaptés à chacune de ces techniques et aux données expérimentales qui leur sont associées. Le tableau 1.7 fait un état des lieux non exhaustif de ressources non spécifiques du riz mais qui se révèlent très utiles pour l'analyse. Par exemple la comparaison de séquences et la recherche d'orthologues avec le génome d'*Arabidopsis* permet de transférer de l'information sur le riz, pour lequel les données expérimentales s'avèrent moins nombreuses.
De manière à faire émerger les besoins des biologistes de la communauté travaillant sur le génome du riz, face aux sources de données spécialisées ou généralistes associées à leurs thématiques, nous proposons, dans les deux sections suivantes, deux scénarios d'usage. Le premier scénario explique la démarche communément adoptée par un biologiste qui à partir du numéro d'accès qui identifie un locus donné, va étoffer sa connaissance par toutes les informations disponibles sur le ou les gènes présents sur ce locus. Le second scénario décrit une démarche classique dès lors qu'il s'agit de réaliser des comparaisons inter-génomes.

1.4.1 Recherche d'information en génomique fonctionnelle

L'exemple retenu porte sur le locus du génome du riz identifié par le numéro d'accès *Os09g33930.1* dans la source de données OryGenesDB[6]. La consultation d'OryGenesDB donne une première série d'informations associées à ce locus (séquence nucléotidique, gène(s), chromosome, ...) (figure 1.8). La fonction biochimique retournée pour la protéine traduite à partir du gène associé au locus *Os09g33930.1* correspond à la fonction enzymatique « farnesyltransferase ».
L'annotation *ab initio* du TIGR ainsi que l'information associée aux molécules d'ADNc (obtenues à partir des transcrits) indiquent trois formes d'épissage alternatif pour le gène considéré. Plusieurs points de vue informationnels sont également disponibles. La couche d'information *SSH (subtractive suppression hybridization) salt* indique une sur-expression de ce gène dans les racines en condition de stress salin. La couche d'information *FST (Flanking sequence tag)* montre qu'il existe des lignées de riz ayant incorporé un élément T-DNA dans ce gène, et donc qu'il existe potentiellement des mutants. Afin d'avoir des informations sur le rôle de ce gène au niveau physiologique, le biologiste va chercher à partir de la couche d'information FST, (par exemple DAL6F01)(figure 1.9) l'existence d'observations sur un phénotype mutant. Le détail du lien DAL6F01 lui donne les caractéristiques de la séquence et le nom de la lignée mutante observée. Il va alors s'adresser à la base de données Oryza Tag Line[7] (figure 1.10) en interrogeant à partir du nom de la lignée et trouver que les plantes mutantes associées à ces lignées mutantes ont des feuilles enroulées et horizontales, de petite taille. Pour confirmer la relation entre le gène étudié et le phénotype trouvé par consultation d'OrygeneDB et d'Oryza Tag Line, le biologiste a alors le loisir d'interroger de nouvelles sources de données chevauchantes (autres collections de mutants par exemple) ou bien de rechercher des informations complémentaires comme par exemple la recherche de gènes orthologues chez *Arabidopsis* dont les gènes sont très précisément décrits sur le portail TAIR.
De manière plus détaillée, la base de données GreenPhyl[8] fournit des données d'orthologie

[6] http ://orygenesdb.cirad.fr
[7] http ://urgi.versailles.inra.fr/OryzaTagLine/

Database	Description	URL
RED	Rice Expression Database	http://red.dna.affrc.go.jp/RED/
RMOS	Rice Microarray Opening Site	http://cdna01.dna.affrc.go.jp/RMOS/
Rice Array Db	NSF Rice Oligonucleotide Array Project	http://www.ricearray.org/
Yale Plant Genomics	Gene expression from tiling path arrays	http://plantgenomics.biology.yale.edu/
Rice MPSS	Rice Massive Parallel Signature Sequencing gene expression database	http://mpss.udel.edu/rice/

FIG. 1.6 – Bases de données de transcriptome (adapté de Antonio et al. [ABY+07])

Database	Description	URL
PlexDB	Plant Expression Database	http://www.plexdb.org/
GRIN	Plant germplasn resources information network	http://www.ars-grin.gov/
TAIR	The *Arabidopsis* Information Resource	http://www.arabidopsis.org/
PLACE db	Plant *cis*-acting regulatory DNA elements	http://www.dna.affrc.go.jp/PLACE/
PlantCARE	Plant *cis*-acting regulatory DNA elements	http://bioinformatics.psb.ugent.be/webtools/plantcare/html/
MATDB	MIPS *Arabidopsis thaliana* database	http://mips.gsf.de/proj/thal/db/
Plant Genomes Central	NCBI Plant Genomes Central – genome projects in progress	http://www.ncbi.nlm.nih.gov/genomes/PLANTS/PlantList.html
EXPASY	Index to other plant specific databases	http://www.expasy.org/links.html

FIG. 1.7 – Autres bases de données en génomique fonctionnelle (adapté de Antonio et al. [ABY+07])

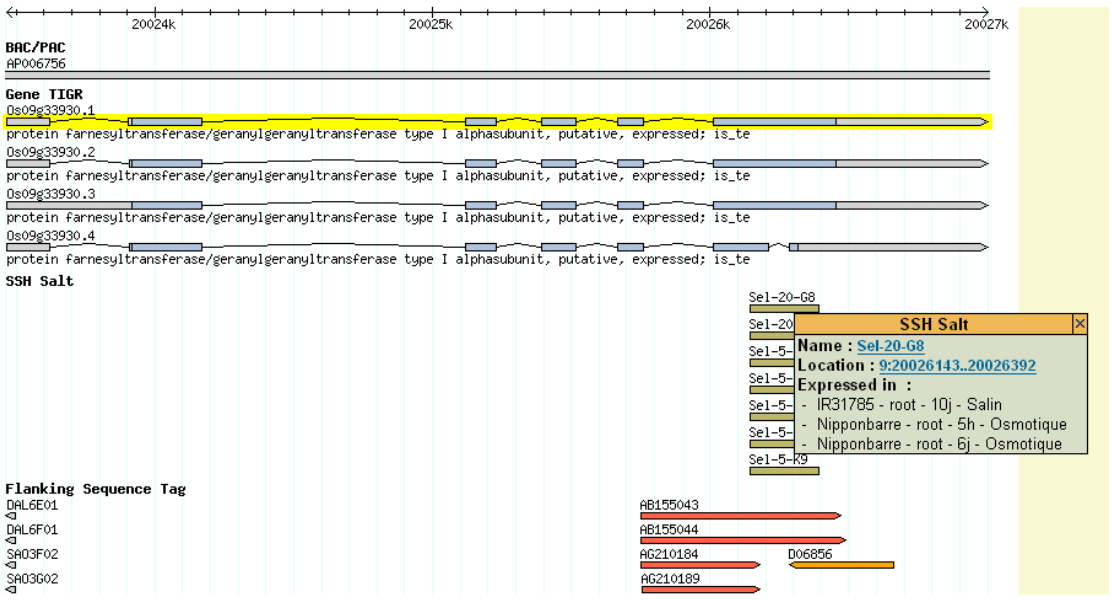


FIG. 1.8 – Copie d'écran du locus Os09g33930.1 avec ses annotations

Synthèse des résultats présents dans la base de données pour DAL6F01

Nom de la FST	**DAL6F01**
Nom de la Plante	**AMTD01**
Type de TDNA	L
Transformation	4
Construit	P4978
Date d'insertion	22-SEP-03
Longueur de la séquence nettoyée	393 pb (Pourcentage de GC : 36%)
Longueur de la séquence brute	572 pb

Sequence Nettoyée

```
>DAL6F01
TTGGTTGCTTCTTCAACGGCTGGATGGTGATGACTGCGCTGCTTTGACTT
GCAGTCAAGGCTGTAACTTCAGCCTATATCTGCACCCACTTGTCATGGAC
TCAATGGGTTCCATATAACATTGTCACAGGGTGATAGTAACAAACTAAAG
AGTACAAAAAGAGTAATCAGCAGAATTCAGAAACGAACTTAAAAACAGTG
CAGGATTCAGGTTTTGGTCCTAATTACTCCATAGGATTCAGGTTTTGGTC
CTAAATACTCCATAGATTATTTACTTCAAATATAAATTACTCCTTCCATC
CCAAAATATACGTTTGCTAGTAGTATGTTTTTTCCAAAGAACTTTTATCT
ATTAGATTGCTTACACCACTTATAACATATTTGTTAATTTTTT
```

FIG. 1.9 – Détail de la séquence FST DAL6F01. La fiche donne notamment la correspondance avec la lignée mutante.

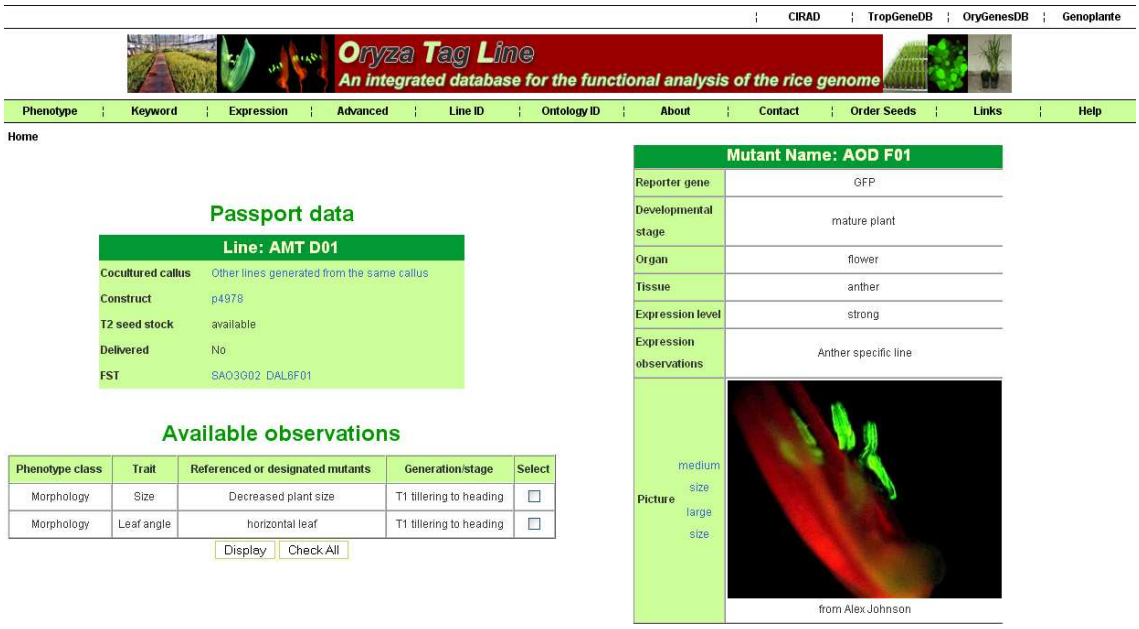
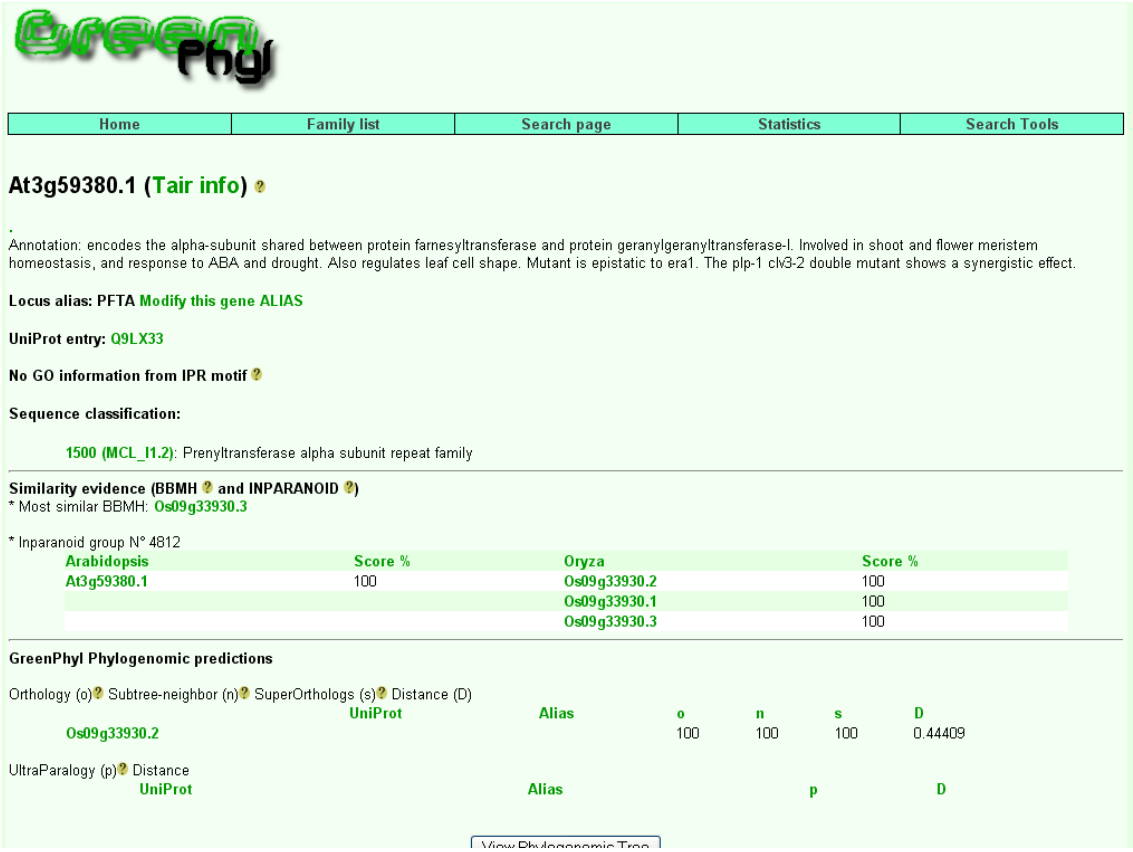


FIG. 1.10 – Page d'observations phénotypiques pour la lignée AMT D01.

entre les gènes de riz et les gènes d'*Arabidopsis*. A partir du nom désignant le locus du riz considéré (Os09g33930.1), le biologiste identifie alors trois variants de gène chez le riz qui ont un seul orthologue chez *Arabidopsis* (AT3G59380.1). L'information disponible sur le site du TAIR[9] [GHBC+02] à ce sujet apporte des renseignements complémentaires (figure 1.12). La description valide la fonction biochimique trouvée précédemment et précise que l'expression de la fonction s'effectue dans les racines et dans les feuilles, en réponse à des phénomènes de stress tels que la sécheresse ou l'application d'ABA. Chez Arabidopsis, le gène AT3G59380.1 contrôle également la forme de la feuille. Les annotations réalisées à l'aide du vocabulaire contrôlé Gene Ontology confirment les observations précédentes. Les titres des publications disponibles dans la source de données de références bibliographiques PubMed à ce sujet, indiquent clairement une relation entre l'enzyme farnesyltransférase et la réponse précoce aux stress hydriques. Ce scénario fait état de la consultation successive de quatre sources de données pour pouvoir, à partir du nom d'un locus précis situé sur le chromosome 9 du génome du riz, enrichir sa connaissance sur le gène associé et attacher une fonction biologique à la protéine codée. Ces sources de données sont par ailleurs des sources à visée intégrative, c'est à dire qu'elles sont organisées pour offrir une vue synthétique combinant différents points de vue ou différents résultats expérimentaux (associés à différentes techniques expérimentales) rendant ainsi les choses moins difficiles à l'utilisateur. Il n'en reste pas moins la nécessité pour l'utilisateur de connaître l'existence et de disposer de ces sources ainsi que la nécessité de savoir les utiliser. La figure 1.10 propose un diagramme de séquence UML qui reprend l'ensemble des activités décrites dans le scénario.

[8] http ://greenphyl.cines.fr
[9] http ://www.tair.org



Home | Family list | Search page | Statistics | Search Tools

At3g59380.1 (Tair info)

Annotation: encodes the alpha-subunit shared between protein farnesyltransferase and protein geranylgeranyltransferase-I. Involved in shoot and flower meristem homeostasis, and response to ABA and drought. Also regulates leaf cell shape. Mutant is epistatic to era1. The plp-1 clv3-2 double mutant shows a synergistic effect.

Locus alias: PFTA Modify this gene ALIAS

UniProt entry: Q9LX33

No GO information from IPR motif

Sequence classification:

1500 (MCL_I1.2): Prenyltransferase alpha subunit repeat family

Similarity evidence (BBMH and INPARANOID)
* Most similar BBMH: Os09g33930.3

* Inparanoid group N° 4812

Arabidopsis	Score %	Oryza	Score %
At3g59380.1	100	Os09g33930.2	100
		Os09g33930.1	100
		Os09g33930.3	100

GreenPhyl Phylogenomic predictions

Orthology (o) Subtree-neighbor (n) SuperOrthologs (s) Distance (D)

UniProt	Alias	o	n	s	D
Os09g33930.2		100	100	100	0.44409

UltraParalogy (p) Distance

UniProt	Alias	p	D

View Phylogenomic Tree

FIG. 1.11 – Recherche des gènes orthologues au locus Os09g33930.1 dans greenphyl

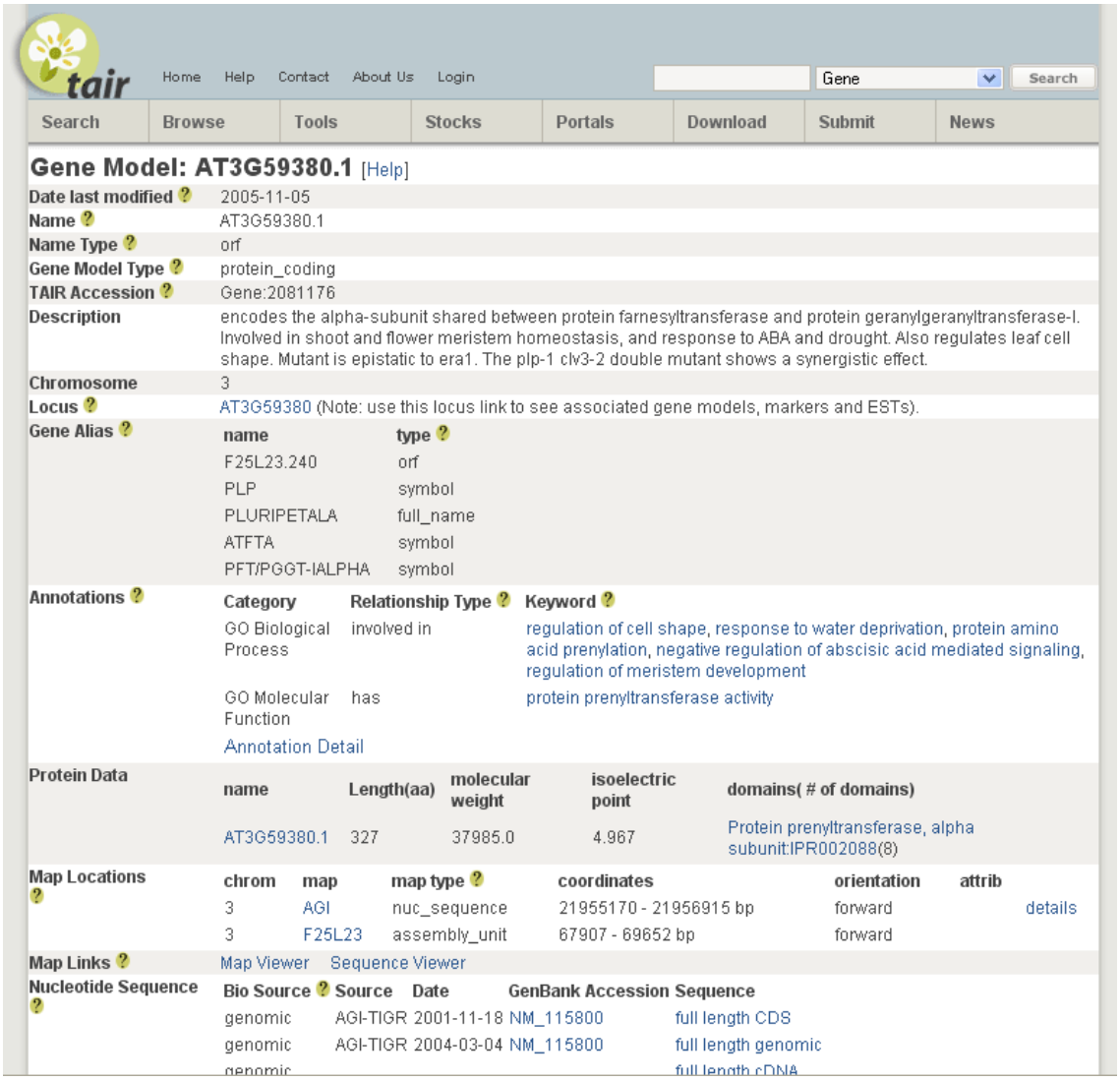


FIG. 1.12 – Fiche Tair d'annotation correspondant au gène AT3G59380.1 orthologue des gènes de riz

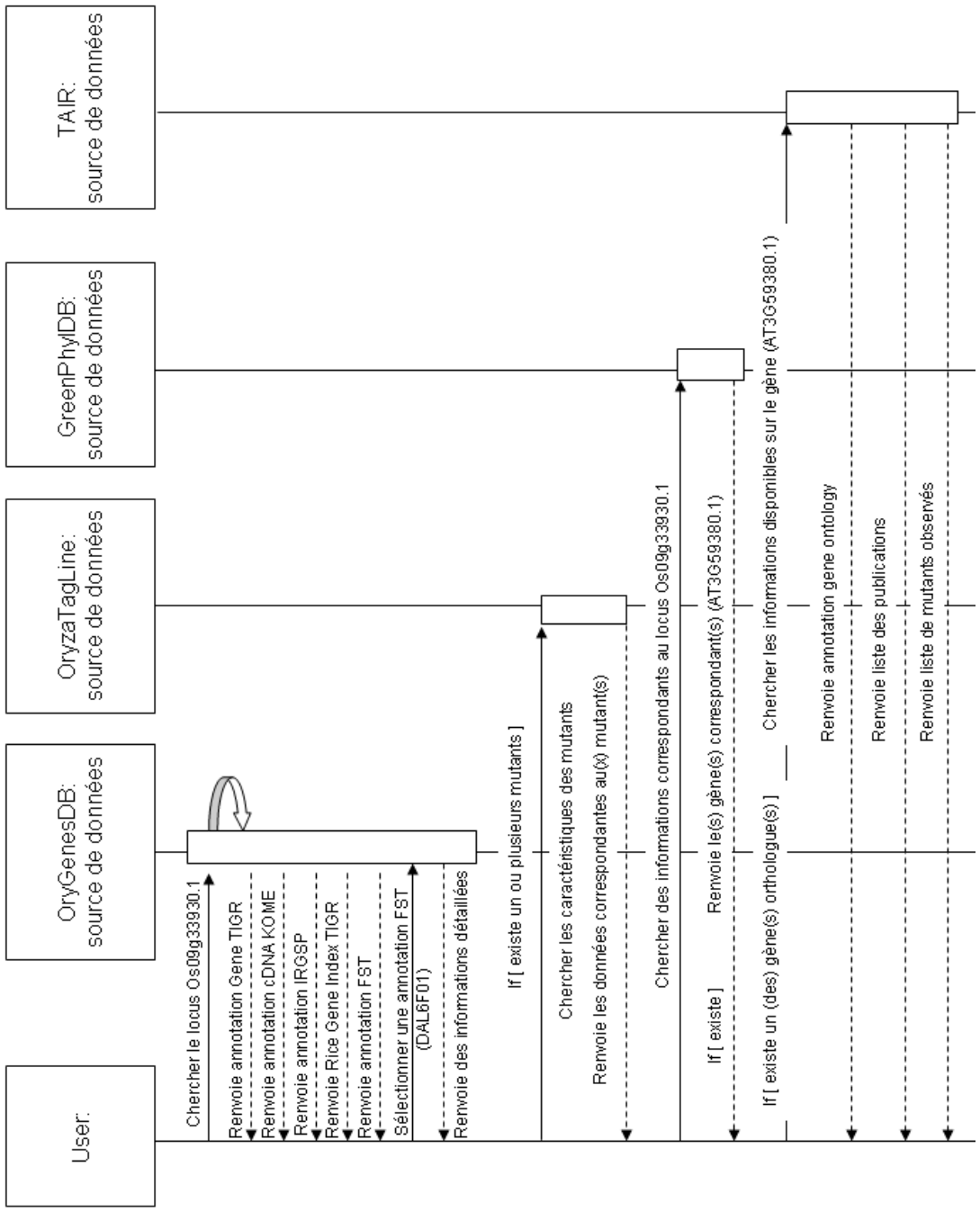


FIG. 1.13 – Diagramme de séquence représentant l'accès aux différentes sources

1.4.2 Exploitation des relations de synténie pour la découverte de gène

Lorsque le génome d'une espèce n'a pas fait l'objet d'expérimentations suffisantes pour en identifier la totalité du répertoire génique ou encore pour identifier pour chacun des gènes considérés, les fonctions biologiques sous-jacentes, le chercheur a alors l'opportunité de s'appuyer sur des données génomiques provenant d'espèces voisines ayant fait l'objet d'études plus poussées. Les deux scénarios d'usage présentés ci-dessous, illustrent, à cet effet, l'exploitation faite par les biologistes des relations de synténie entre espèces apparentées afin de faire émerger de nouvelles connaissances sur l'espèce la moins bien étudiée et notamment pour dégager des informations faisant le lien entre le génotype et le phénotype de l'espèce en question.

1.4.2.1 Recherche d'un gène candidat

Un généticien étudiant le sorgho a caractérisé, par le biais d'analyses statistiques, des marqueurs RFLP liés à un QTL de résistance à la sécheresse. Il s'interroge ensuite sur l'identité des gènes candidats qui sous-tendent ce QTL. Dans ce sens, il exploite les relations de synténie qui co-existent dans le génome des céréales ; dans notre scénario, il va s'attacher à exploiter, les informations génomiques attachées au riz, dans son contexte d'étude à savoir le génome du sorgho.

- Dans un premier temps, le travail va consister à vérifier l'existence d'une séquence biologique correspondant à ces marqueurs dans les bases de données spécifiques des céréales (e.g. Gramene ou GrainGene). Si le biologiste trouve un numéro d'accession GenBank pour les deux marqueurs, il va alors faire l'acquisition des séquences associées, au format Fasta grâce aux liens de références croisées portant sur les numéros d'accessions et par navigation entre Gramene et GenBank. Les séquences sont ensuite positionnées sur les pseudo-molécules de riz en exploitant les outils (visualisateur de génomes, outil de recherche de similarité) de la base de données OryGenesDB. Si les deux séquences se positionnent sur le même chromosome de riz à des distances physiques proches, il est probable que la synténie soit bien conservée dans cette région. Le généticien peut extraire, via un autre outil d'OryGenesDB, tous les gènes compris entre les positions des deux marqueurs ainsi que leurs annotations. Chaque gène, dans cet intervalle, sera étudié afin de déterminer ses implications potentielles au regard de la fonction physiologique observée.
- Si par contre, il n'existe pas de correspondance en terme de séquences dans les bases de données spécifiques des céréales, pour les marqueurs considérés ; le chercheur va alors s'employer à trouver des marqueurs apparentés pour lesquels une séquence est connue. A cet effet, il exploite les cartes génétiques proposées par Gramene et notamment il exploite les cartes qui comportent des marqueurs à large spectre d'hybridation. Ces marqueurs sont positionnés sur des cartes génétiques de plusieurs espèces proches et peuvent alors servir de passerelles informationnelles entre espèces. Ces cartes génétiques vont ainsi lui permettre de définir la position relative des marqueurs sorgho sur les cartes du riz à partir d'interpolations sur les distances génétiques et au prix probablement d'une certaine perte de précision. Il lui reste alors à définir les marqueurs riz les plus proches du QTL d'intérêt. Une fois ces marqueurs identifiés, le processus d'extraction de l'information sur la zone génomique localisée se ramènera au processus vu précédemment.

La base de données OryGenesDB permet ensuite de vérifier l'existence de mutants associés au(x) gène(s) candidat(s) dans l'une des collections de mutants d'insertion. Si un mutant ou bien encore des mutants sont trouvés, il reste alors à chercher dans la base de données Oryza Tag Line la présence de phénotype(s) particulier(s) qui correspondent à cette ou à ces mutations. Le fait de trouver par exemple de nombreuses données en corrélation va conduire le biologiste à renforcer ses convictions au sujet de l'importance de son gène d'intérêt. Il peut alors lancer une série de nouvelles expérimentations à la paillasse qui vont de fait être bien ciblées.

Le scénario proposé fait l'usage de plusieurs bases de données et de différents services de traitement proposés par ces mes bases de données. Certaines des opérations de consultation se font par de la navigation entre les bases de données au travers de liens déjà mis en place. D'autres opérations de consultation supposent des accès distincts aux différents sources de données (pas de connexion directe entre GenBank et OryGenesDB par exemple). La figure 1.14 schématise les étapes effectuées par le généticien au travers d'un diagramme de séquence UML.

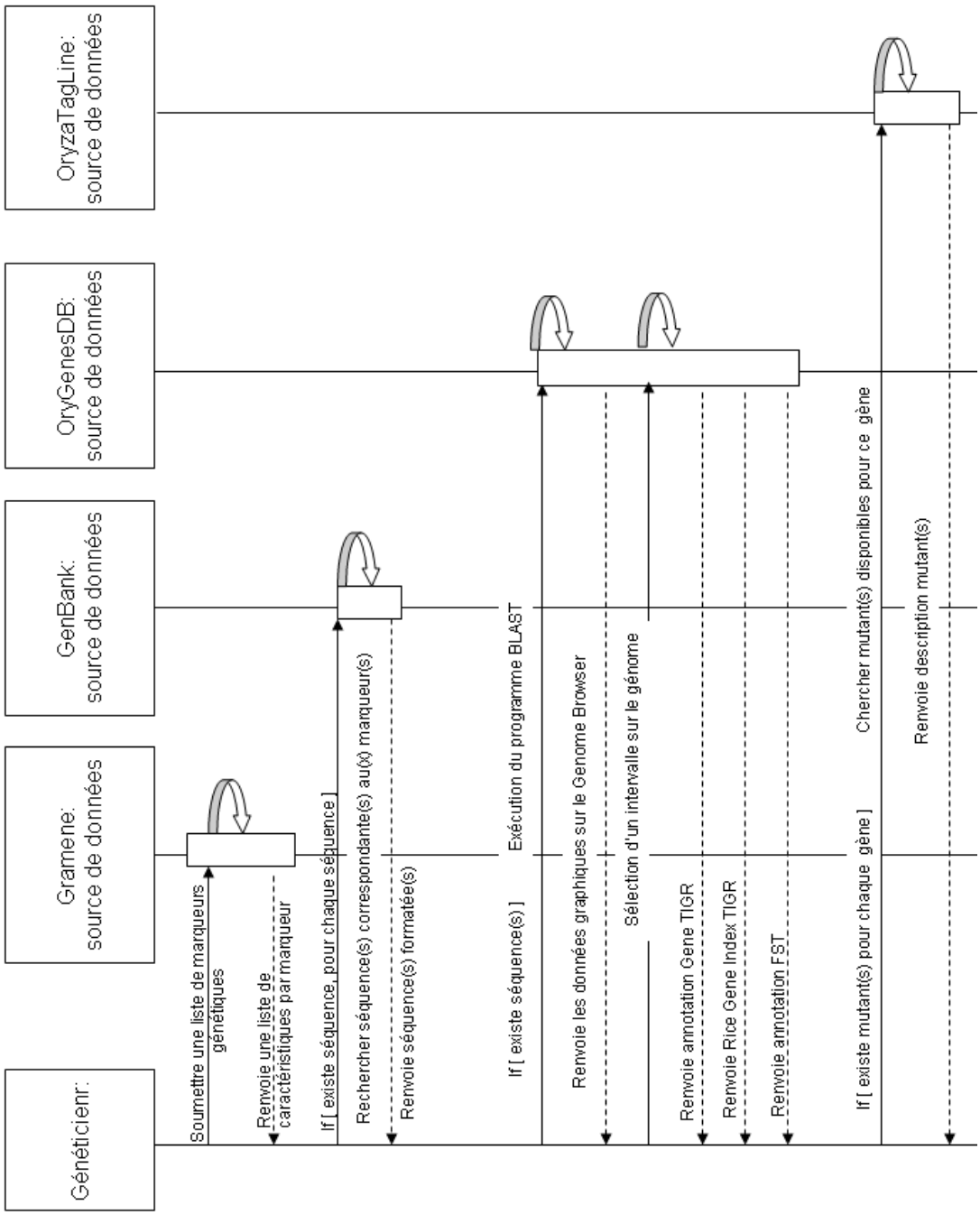


FIG. 1.14 – Diagramme de séquence de la recherche de gènes candidats

1.4.2.2 Détection d'allèles correspondant à un gène candidat

Le même généticien, travaillant sur le génome du sorgho, se concentre maintenant sur l'étude d'un gène identifié chez *Arabidopsis* comme jouant un rôle essentiel dans la croissance racinaire. Il cherche à caractériser dans le génome du sorgho le gène et ses possibles allèles qui vont se révéler être les corollaires du gène d'intérêt chez Arabidopsis. A cet effet, il va mener des expérimentations sur le polymorphisme moléculaire attaché à ce gène et sur les effets de ce polymorphisme sur de possibles variations de croissance racinaire chez le sorgho.

- La première étape va alors consister à rechercher l'orthologue de ce gène chez le sorgho.

    1. une première démarche consiste à utiliser Greenphyl, une ressource qui permet d'identifier l'orthologue chez le riz d'un gène d'*Arabidopsis* par phylogénomique à partir de la confrontation des séquences protéiques associées. Une fois ce gène identifié (s'il en existe un et qu'il est unique), la séquence nucléotidique du gène du riz peut être exploitée dans un second temps, pour rechercher dans la source de données GenBank une séquence de sorgho montrant une forte similarité.

    2. Une deuxième démarche peut consister à révèler directement dans GenBank une similarité entre les séquences de sorgho et la séquence d'intérêt d'*Arabidopsis*. Mais cette démarche peut aboutir à des résultats moins pertinents en sachant que GenBank ne contient qu'une fraction des gènes du génome du sorgho et que les génomes du sorgho et d'Arabidopsis sont respectivement moins proches que les génomes du sorgho et du riz.

- Si lors d'une des deux démarches précédentes, le généticien trouve une séquence sorgho de qualité jugée suffisante, il peut sélectionner des amorces dans la séquence résultat, au moyen d'un outil publique de conception d'amorces de PCR comme Primer 3, et passer ensuite au travail de laboratoire pour réaliser le séquençage nécessaire afin de dégager des éléments de réponse à ses questions. Son objectif est, ici, de se concentrer sur les mutations qui vont avoir une incidence d'un point de vue fonctionnel. Il fera dans ce sens l'acquisition des informations sur les sous-régions fonctionnelles (ou "features" du gène) associées à la séquence résultat. Il va ainsi connaître les zones de bordure (zones d'épissage) entre les introns et les exons afin de pour pouvoir ancrer ses amorces dans les exons les mieux conservés.
- Si aucune séquence de sorgho ne répond aux critères de sélection, une alternative est alors de rechercher les régions les mieux conservées du gène d'*Arabidopsis* afin de pouvoir identifier le gène orthologue présent chez le sorgho à l'aide d'amorces. Dans un premier temps, il faut extraire de GenBank toutes les séquences de gènes complets de céréales (maïs, blé, orge, mil, larmes de Job, etc.) présentant une forte similarité avec le gène du riz, puis réaliser par exemple un alignement multiple au travers par exemple de l'outil en ligne ClustalW du portail SRS et enfin définir des amorces dégénérées permettant d'amplifier le gène chez le sorgho avec un outil comme Oligo6 ou Codehop. Pour ce faire, le généticien va chercher à ancrer ses amorces dans une zone bien conservée. Dans cette perspective, il traduit d'abord la séquence nucléotidique en séquence protéique en utilisant un outil de traduction en ligne et consulte ensuite la signature des domaines fonctionnels présents dans la séquence protéique résultat avec un outil comme ScanProsite ou SignalScan. Il lui reste alors à recadrer les domaines fonctionnels sur la séquence nucléotidique.

Là-encore, le scénario démontre la nécessité de l'accès à différentes ressources (sources de données et outils de traitement) partagés par la communauté. Un diagramme de séquences UML modélise les différentes activités du généticien.

1.4.3 Conclusion sur les scénarios d'usage

Les scénarios d'usage présentés ont pour mérite de proposer différents canevas d'utilisation des sources de données spécialisées et généralistes du domaine. Dans le premier scénario qui est le plus simple, il s'agit essentiellement de consulter différentes sources de données, qui pour certaines d'entre elles, proposent une vision synthétique de l'information. Dans le second scénario, l'usager manipule non seulement différentes sources de données mais aussi des outils de traitement pour mener à bien son étude. Il apparaît donc comme nécessaire pour le biologiste de connaître et aussi de maîtriser de multiples ressources (sources de données comme outils de traitement). Enfin dans le troisième scénario, l'usager est attendu de réaliser à la fois des expérimentations à la paillasse et des expérimentations dites sèches à l'aide des ressources bioinformatiques déjà citées.

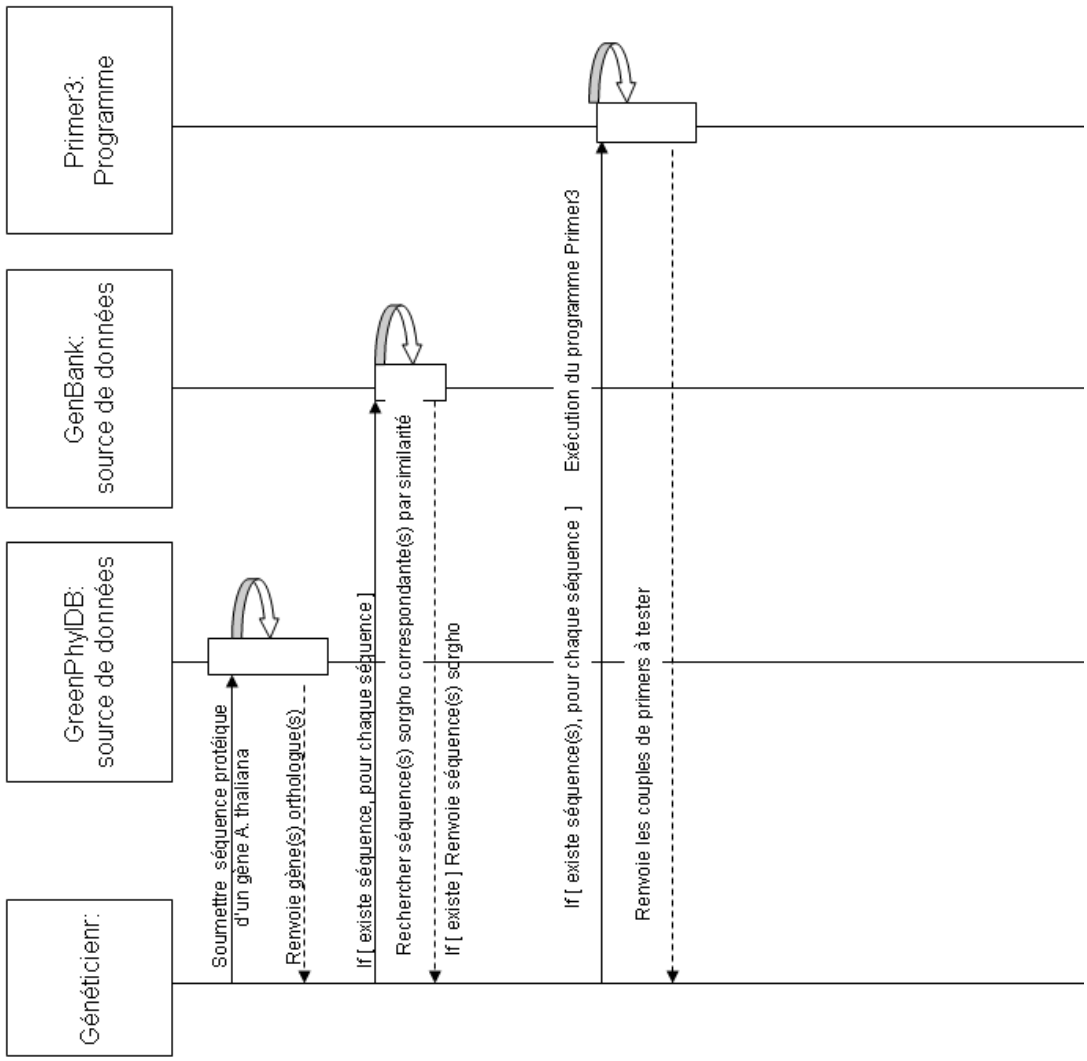


FIG. 1.15 – Diagramme de séquence sur la détection d'allèles

Chapitre 2

Formalismes et modèles des sources

Sommaire

2.1 Partage de l'information biologique

DANS le domaine biologique, l'observation joue un rôle important dans la compréhension des systèmes. L'étude des mécanismes biologiques se fait, en effet, par le biais d'hypothèses, qui le plus souvent, sont basées sur les connaissances acquises sur des modèles similaires. Prenons l'exemple des espèces végétales du riz (*Oryza sativa*) et de l'arabette des dames (*Arabidopsis thaliana*). Nous allons, ainsi, pouvoir transférer des connaissances acquises sur l'espèce la plus étudiée en génomique fonctionnelle, en l'occurrence *Arabidopsis thaliana* vers l'espèce la moins étudiée, ici *Oryza sativa*. Plus précisément, le gène ERECTA (Gene id : At2g26330) est connu chez *Arabidopsis thaliana* pour être impliqué, entre autres, dans les mécanismes de résistance à un pathogène *Ralstonia solanacearum*. La bactérie *Ralstonia solanacearum* est responsable de flétrissement et s'attaque à un large spectre d'hôtes (plus de 200 espèces végétales). La démarche communautaire, qui consiste à désigner des organismes comme *Arabidopsis thaliana* ou *Ralstonia solanacearum* comme étant des organismes modèles, prend ici tout son sens. Les études approfondies menées chez ces organismes vont pouvoir servir de support à de nouvelles études menées chez d'autres organismes et surtout à pouvoir cibler très rapidement les études qui vont se révéler pertinentes pour faire progresser les connaissances. Pour ce qui concerne *Oryza sativa*, un gène potentiellement orthologue du gène ERECTA (entrée Os06g10230 dans OryGenesDB) a été identifié, avec pour toute information disponible, sa séquence nucléique annotée. Il va s'agir alors de réutiliser les connaissances acquises en génomique fonctionnelle autour d'ERECTA (caractérisation de la variabilité d'expression, localisation spatiale et temporelle de l'expression du gène, gènes co-exprimés, etc) et de *Ralstonia solanacearum*, en supposant que ces connaissances sont valides dans le contexte d'*Oryza sativa*.

L'échange et le partage des connaissances ainsi que les vecteurs de communication associés, se révèlent essentiels, non seulement pour comparer des données biologiques provenant de différents organismes et en inférer de nouvelles, mais aussi pour comprendre des phénomènes biologiques de manière globale. La notion de partage est pensée ici de manière large, comme une mise en commun de démarches scientifiques, de données, de modèles de données et d'outils de traitement. Cette notion est, pour beaucoup, à l'origine des besoins d'intégration des sources de données en biologie, qui seront détaillés dans le chapitre 3.

La priorité dans ce chapitre, est toutefois donnée à la mise à disposition, à la fois facilitée et fiable, de données et de modèles de données depuis différents systèmes vers l'ensemble des acteurs de la communauté biologique. Nous discuterons dans une première partie des politiques internationales de partage de données depuis trois décennies et de leurs traductions en terme non seulement de sources de données mises en place mais aussi de moyens mis en oeuvre pour le partage. Nous aborderons dans une deuxième partie, les challenges actuels découlant de ces politiques de partage, à savoir les problèmes posés par l'existence de multiples sources de

données et l'intégration nécessaire de ces sources de données. Enfin, une troisième et dernière partie s'attache à définir des concepts clés tels que les ontologies et les métadonnées se révélant d'importance dans la mise en place de systèmes intégrés pouvant apporter des éléments de réponses aux besoins complexes des biologistes en termes d'analyse et de confrontation des données.

2.1.1 Organisation des sources de données

Données et connaissances scientifiques ont tout d'abord été partagées au travers de publications scientifiques ou encore de conférences orales. La publication, notamment s'est avérée et s'avère encore un vecteur de communication prépondérant dans la mesure ou elle véhicule non seulement les données biologiques et les expertises sur ces données, mais aussi les méthodes expérimentales (véritables recettes de cuisine) utilisées par les auteurs.

Le partage de l'information est un point essentiel pour la communauté biologique permettant ainsi de confronter les connaissances et d'en inférer de nouvelles afin de construire modèles et hypothèses biologiques. L'information a d'abord été véhiculée au travers des publications scientifiques. Ce vecteur demeure, aujourd'hui, un canal de communication très important. Il constitue, d'abord, un outil de promotion des travaux des auteurs, il forme ensuite un tout dans la mesure où il livre les données, les expertises des auteurs mais également la manière dont les expériences ont été réalisées autorisant d'autres scientifiques à les reproduire. Avec les réseaux informatiques et Internet, de nouveaux vecteurs sont apparus, les données expérimentales sont alors collectées, gérées et véhiculées de manière séparée des publications afin d'en faciliter la consultation mais aussi le traitement.

Ainsi, Margaret Dayhoff propose en 1965 un atlas sur les protéines, leurs structures et leurs séquences [DEP72]. En 1971, l'atlas devient la première banque de donnée consacrée aux molécules biologiques. Nommée Protein Data Bank[10], elle rassemble, au niveau mondial, les structures tridimensionnelles (ou structures 3D) de macromolécules biologiques : essentiellement protéines et acides nucléiques. Les bases de nouveaux besoins et de nouvelles disciplines étaient posées. En effet, la découverte de techniques de biologie moléculaire telles que la PCR[11] et le séquençage à haut débit, marquent une rupture par rapport aux traitements des données tels qu'ils étaient réalisés jusqu'alors. Ces techniques ont augmenté considérablement le nombre de séquences nucléotidiques produites, faisant émerger de nouveaux besoins en matière de stockage, de gestion et de diffusion de l'information. Il y a dès lors, une volonté politique de créer des organismes capables de centraliser et de distribuer les données produites au niveau mondial ainsi que de former les scientifiques aux disciplines et technologies émergentes. En Europe, l'EMBO[12], une organisation créée en 1962, obtient le soutien des instances politiques des principaux pays européens afin de promouvoir la recherche publique en biologie. L'EMBO permet ainsi la création en 1974, du laboratoire européen EMBL (the European Molecular Biology Laboratory), et appuie fortement le développement, dès 1986, de la première banque de séquences nucléiques : l'EMBL data library ([HC86, KAA+07]) qui est maintenue depuis par EMBL et distribuée publiquement aux biologistes du monde entier. Depuis sa création, EMBL

[10] http ://www.rcsb.org/pdb
[11] Polymerase Chain Reaction
[12] European Molecular Biology Organisation

joue un rôle d'appui dans la formation des scientifiques et la mise en oeuvre de démarches et de technologies instrumentalisant les sciences de la vie. Dans ce sens, EMBL a supporté l'essor de la bioinformatique en Europe, en facilitant la mise en place de nombreuses bases de données dédiées à un organisme ou encore à une famille de molécules, et l'automatisation de traitements accompagnant les projets de séquençage d'organismes modèles.

Grâce à des volontés politiques de centralisation et de partage de la connaissance, à l'image d'EMBL Data Library, de nombreuses sources de données (banques ou bases de données) ont été développées sous l'impulsion d'organismes publics gouvernementaux. Les centres de ressources et les principaux journaux scientifiques du domaine collaborent afin d'inciter les biologistes à contribuer à l'alimentation des bases de données. Prenons l'exemple des séquences nucléiques, pour soumettre un article mettant en lumière de nouvelles séquences, les scientifiques doivent au préalable soumettre leurs données dans une des sources de données publiques de séquences nucléiques et fournir en retour au journal les identifiants des séquences attribués par la source de données. Le gain pour le lecteur de la publication est indéniable, il peut ainsi, en complément de la publication, avoir accès aux informations associées contenues dans la source de données.

Des politiques de collaboration se sont également instaurées entre les différents centres de ressources. Prenons l'exemple, toujours pour les données de séquences nucléiques, de l'EMBL data library ([HC86, KAA+07]), de GenBank ([BFG+85, BBF+86, BKML+06]) et de DDBJ (DNA Data Bank Japan) qui centralisent les données respectivement en Europe, Etats Unis et Japon et les publient internationalement et de manière publique. L'objectif entre ces centres de ressources est de donner accès à la même information, quel que soit, d'une part, le point d'entrée de consultation choisi par l'usager et quel que soit, d'autre part, le centre de ressource choisi par le biologiste pour soumettre ses données de séquence. A cet effet, les trois centres ont défini un schéma de base de données ainsi qu'un format de soumission de données communs. Ils ont également adopté une politique de mise à jour synchronisée afin d'offrir à leurs usagers strictement la même information. En 1997, afin d'officialiser leur collaboration, EMBL Data Library, GenBank et DDBJ se sont regroupés dans le consortium INSDC[13] (International Nucleotide Sequence Database Collaboration). Les sources de données consacrées aux séquences protéiques ont, depuis, adopté la même démarche et se sont regroupées, en 2002, dans le consortium UniProt[14] [BAW+05, LDB+04, ABW+04].

Parallèlement aux bases de données dites généralistes qui centralisent au niveau mondial la production de données biologiques, la dernière décennie a vu le développement de bases de données dites spécialisées. Leur développement est lié à des besoins spécifiques en terme d'intégration de données, de filtrage d'information ou d'outils d'analyses dédiés. Elles ont un rôle complémentaire des sources de données généralistes en enrichissant l'information produite par des expertises supplémentaires. Par exemple, l'institut TIGR[15] met à disposition des résultats d'analyses bioinformatiques effectuées (e.g. annotations) sur les séquences de BAC de riz[16] soumis dans le consortium INSDC. Des politiques de collaboration existent également entre les

[13] http ://www.insdc.org
[14] Universal Protein Resource
[15] www.tigr.org
[16] http ://www.tigr.org/tdb/e2k1/osa1/

bases de données généralistes et spécialisées. EMBL a mis en place des partenariats avec des sources spécialisées (e.g. IMG-T, e.Coli) pour la validation et l'annotation des séquences qui leur sont soumises.

Le regroupement thématique des données (séquences nucléiques, séquences protéiques, structure tridimensionnelle des molécules, cartes génétiques, etc), leur interdépendance et enfin les différentes politiques de collaboration ont fait émerger la notion de référence croisée. Une référence croisée résulte simplement de la concaténation de l'identifiant de la donnée considérée et du nom de la source de données qui la gère. Par exemple, CDD :66084 correspond au domaine fonctionnel "Glucan Synthase" identifié par le numéro d'accès 66084 dans la source de données dédiée aux domaines conservés CDD. Le partage de l'information entre les sources s'effectue tout simplement par des référencements de part et d'autre dans les sources de données, via ces références croisées. Au niveau des portails Web Entrez et SRS, les références croisées vont œuvrer à la manière des liens hypertextes, il est ainsi rendu possible de "naviguer" depuis une séquence nucléotidique vers d'autres types d'informations qui lui sont associées comme le taxon d'origine, la ou les séquences protéiques qui en résultent, les références bibliographiques, etc.

L'augmentation du nombre de sources de données est à la hauteur de l'explosion du volume des données, il est donc difficile d'en avoir une vision synthétique. Le journal *Nucleic Acid Research* (NAR) [Gal07] réalise un inventaire annuel non exhaustif des sources de données disponibles dans un numéro spécial consacré aux bases de données du domaine. En 2007, il dénombrait ainsi 968 bases de données qu'il organisait en 14 catégories. Les catégories définies, sont susceptibles d'évoluer dans le temps, elles ont surtout le mérite d'organiser les sources de données entre elles. Pour exemple, la catégorie " Plant databases " s'est, au fil du temps, scindée en plusieurs sous-catégories traitant des principales espèces végétales comme le riz ou Arabidopsis thaliana. Pour illustration, la figure 2.1 présente un extrait de ce catalogue de bases de données. Les catégories sont organisées, soit selon la nature des objets biologiques (ADN, ARN, protéine, organelle, cellule, tissu, organe, organisme), soit selon différents points de vue sur ces objets biologiques (séquence, structure 3D, cartes génétiques, taxonomies), soit selon différentes techniques d'obtention (micro array, macro array, SAGE), soi selon différentes thématiques d'étude (voies métaboliques , protéomique, système immunitaire).

Catégorie	Exemple	Nom ou Description	URL
Nucleotide Sequence Databases	DDBJ	DNA Data Bank of Japan	http://www.ddbj.nig.ac.jp
	EMBL	The EMBL Nucleotide Sequence Database	http://www.ebi.ac.uk/embl/
	GenBank	NCBI nucleotide Sequence Database	http://www.ncbi.nlm.nih.gov/
RNA sequence databases	Plant snoRNA DB	Base de snoRNA	http://www.ncbi.nlm.nih.gov/
Protein sequence databases	PIR-PSD	Base de séquences protéiques	http://pir.georgetown.edu
	Swiss-Prot (UniProtKB\Swiss-Prot)	Base intégrative de séquences protéiques	http://www.expasy.org/sprot
	UniProt	Ressources de séquences protéiques	http://www.uniprot.org
	PROSITE	Motifs protéiques	http://www.expasy.org/prosite
	InterPro	Domaines protéiques	http://www.ebi.ac.uk/interpro
	Pfam	Domaines protéiques	http://www.sanger.ac.uk/Software/Pfam/
Structure Databases	PDB	Structures 3D protéiques	http://www.rcsb.org/pdb/
Genomics Databases (non-vertebrate)	TIGR Gene Indices	Collection de bases de données de séquences	http://www.tigr.org/tdb/tgi/
	KEGG	Kyoto Encyclopedia of Genes and Genomes	http://www.genome.ad.jp/kegg
	Entrez Genomes	Collection de bases de données de séquence du NCBI	http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Genome
Metabolic and Signaling Pathways	BioCyc	Base de voies métaboliques pour les bactéries	http://biocyc.org/
	MetaCyc	Base de voies métaboliques	http://metacyc.org/
Human and other Vertebrate Genomes	Ensembl	Human genome browser	http://www.ensembl.org/
Human Genes and Diseases	OMIM	Online Mendelian Inheritance in Man	http://www.ncbi.nlm.nih.gov/Omim/
Microarray Data and other Gene Expression Databases	CATMA	Complete Arabidopsis Transcriptome MicroArray	http://www.catma.org
Proteomics Resources	PlantMarkers	Base de marqueurs spécifiques plantes	http://markers.btk.fi
Other Molecular Biology Databases	PubMed	Base de données bibliographique	http://pubmed.gov
Organelle databases	ChloroplastDB	Spécifique des génomes chloroplastiques	http://chloroplast.cbio.psu.edu/
Plant databases	Gramene	Base génétique pour les céréales	http://www.gramene.org
	TropGENE DB	Base génétique spécifique des espèces tropicales	http://tropgenedb.cirad.fr/
	GénoPlante-Info	Plate-forme intégrative pour les données plantes	http://www.genoplante.com/
	FLAGdb++	Base génomique spécifique d'Arabidopsis	http://urgv.evry.inra.fr/projects/FLAGdb++/HTML/index.shtml
	Oryza Tag Line	Base de mutants phénotypiques riz	http://urgi.versailles.inra.fr/OryzaTagLine/
	OryGenesDB	Base génomique spécifique riz	http://orygenesdb.cirad.fr/
Immunological databases	IMGT	the international ImMunoGeneTics information system®	http://imgt.cines.fr

FIG. 2.1 – Extraits du catalogue de bases de données édité par le NAR (adapté de [Gal07])

2.1.2 Les moyens mis en oeuvre pour partager l'information

Au niveau mondial, les centres de ressources, créés au travers des politiques de collaboration, ont défini différents des formats de stockage afin de partager l'information produite par l'ensemble des communautés biologiques. Ces formats sont devenus très rapidement des standards sur lesquels se sont appuyés des outils de traitements pour définir à la fois leurs entrées et leurs sorties.

Pendant longtemps, les fichiers plats dotés d'un format propriétaire ont été la principale forme de stockage de l'information biologique, particulièrement pour ce qui concerne les séquences biologiques. Ainsi, les banques généralistes de séquences (e.g. GenBank, EMBL, PIR, PDB, etc) distribuent leurs données selon leurs propres formats textuels [KFG84]. La figure 2.2 représente une séquence nucléique dans le format EMBL. Ce fichier possède une structure avec des identifiants de lignes à deux lettres qui permettent à des programmes d'en extraire plus facilement de l'information (e.g. AC signifie accession number, KW signifie keywords, etc.). La force de ces formats est d'être facilement lu, interprété et de fait totalement adopté par l'ensemble des scientifiques[17]. Une règle d'organisation simple fait correspondre chaque fichier à un seul objet biologique. Ainsi chaque séquence biologique, et ses informations connexes (sous-régions fonctionnelles, auteurs, publications, références croisées, ...) sera décrite au travers d'un seul fichier. Le recours à des fichiers textuels au format propriétaire pose des problèmes évidents de standardisation. Il n'existe ainsi ni consensus ni modèle défini et partagé pour que des applications puissent traiter les données de manière uniforme. Des langages, de description de données, standards comme ASN.1 puis plus récemment XML sont également utilisés aujourd'hui pour décrire et échanger les données biologiques. Les formats de séquences propriétaires n'en demeurent pas moins encore très largement exploités. Faciles à exporter et échanger, ils sont aussi préférés par les biologistes qui peuvent ainsi travailler directement sur les fichiers correspondants à leurs objets biologiques d'intérêt. Ce type de format valorise également directement les auteurs et leurs travaux en associant leurs noms aux objets biologiques caractérisés et aux fichiers sous-jacents. Enfin, de nombreux logiciels dédiés à l'analyse de séquences biologiques (GCG, FASTA, BLAST, suite EMBOSS, ALIGN, CLUSTAL, etc) ont pris l'habitude de manipuler en entrée, bon nombre de ces formats. Par exemple, le format FASTA est utilisé par la majorité des programmes d'alignement comme SIM ou T-COFFEE et de recherche de similarité de comparaison de séquences comme BLAST [AGM+90] ou encore FASTA.
Depuis quelques années, les efforts de standardisation menés par les consortiums ont conduit à l'apparition de nouveaux formats textuels. Par exemple, le Gene Ontology Consortium, dont l'objectif initial est de centraliser les nomenclatures et les vocabulaires contrôlés en biologie moléculaire, est à l'origine de nombreux formats d'échange dont le format OBO (voir section 2.3.3.1). Le format GFF[18](General Feature Format) est un autre format créé pour structurer l'annotation des séquences et notamment les informations que l'on qualifie de "features" dans la description des séquences d'ADN, d'ARN et protéiques.

Le langage de balisage standard XML (eXtensible Markup Language) apporte progressivement une alternative aux formats de fichier propriétaires. Les grandes banques de séquences comme EMBL proposent aujourd'hui à leurs utilisateurs plusieurs grammaires XML (DTD) décrivant les séquences nucléiques et donnent accès aux fichiers de données respectant ces gram-

[17] http ://www.ebi.ac.uk/2can/tutorials/formats.html
[18] http ://www.sanger.ac.uk/Software/formats/GFF/

```
ID   CL520694; SV 1; linear; genomic DNA; GSS; PLN; 257 BP.
XX
AC   CL520694;
XX
DT   04-APR-2004 (Rel. 79, Created)
DT   26-MAY-2006 (Rel. 87, Last updated, Version 2)
XX
DE   MUL5B09 Flanking Sequence Tag of Oryza sativa T-DNA insertion lines Oryza
DE   sativa (japonica cultivar-group) genomic, genomic survey sequence.
XX
KW   GSS.
XX
OS   Oryza sativa (japonica cultivar-group)
OC   Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
OC   Spermatophyta; Magnoliophyta; Liliopsida; Poales; Poaceae; BEP clade;
OC   Ehrhartoideae; Oryzeae; Oryza.
XX
RN   [1]
RP   1-257
RX   DOI; 10.1111/j.1365-313X.2004.02145.x
RX   PUBMED; 15255873.
RA   Sallaud C., Gay C., Larmande P., Bes M., Piffanelli P., Piegu B., Droc G.,
RA   Regad F., Bourgeois E., Meynard D., Perin C., Sabau X., Ghesquiere A.,
RA   Glaszmann J.C., Delseny M., Guiderdoni E.;
RT   "High throughput T-DNA insertion mutagenesis in rice: a first step towards
RT   in silico reverse genetics.";
RL   Plant J. 39(3):450-464(2004).
XX
CC   Contact: Guiderdoni
CC   UMR PIA Biotrop program
CC   CIRAD
CC   TA 40/03 ave Agropolis 34398 Montpellier cedex 5 FRANCE
CC   Tel: 33467615629
CC   Fax: 33467615605
CC   Email: emmanuel.guiderdoni@cirad.fr
CC   Class: TDNA tagged.
XX
FH   Key             Location/Qualifiers
FH
FT   source           1..257
FT                    /organism="Oryza sativa (japonica cultivar-group)"
FT                    /cultivar="Nipponbare"
FT                    /mol_type="genomic DNA"
FT                    /clone_lib="Flanking Sequence Tag of Oryza sativa T-DNA
FT                    insertion lines"
FT                    /note="PCR was performed on DNA of primary transformants of
FT                    Orysa sativa plants. The DNA fragment(s) resulting of PCR
FT                    were directly sequenced from the left border to determine
FT                    the genomic sequence flanking the insertion. T-DNA derived
FT                    sequences were removed. Information to order the
FT                    corresponding mutant line and a link to a database
FT                    providing a graphical display is available from june 2004
FT                    at http://genoplante-info.infobiogen.fr/oryzatagline/. This
FT                    sequence has been generated in the framework of the French
FT                    plant genomics program Genoplante
FT                    (http://www.genoplante.org and
FT                    http://genoplante-info.infobiogen.fr)."
FT                    /db_xref="taxon:39947"
XX
SQ   Sequence 257 BP; 94 A; 52 C; 46 G; 65 T; 0 other;
     tccaccaaaa tatatgtgca aattcgatct aaacgtccac aaacaaaaaa gaccaattca       60
     gatatataca gtgcactatt atacatacgc tgaatttgtc tgctgtatat ctcaacgtgt      120
     gagttgactt aggactaaag atctagtgaa atagtatata tatatagtat aaataccacc      180
```

FIG. 2.2 – Format EMBL d'une séquence nucléotidique

maires (figure 2.2). La grande force du langage XML est d'être, avant tout, un standard et donc d'être supporté par un ensemble d'applicatifs et d'interfaces de programmation qui vont en faciliter l'exploitation. Les primitives de modélisation proposées par XML (ELEMENT, ENTITY, ATTLIST ...) sont simples mais vont grandement faciliter l'accès et le traitement des données par comparaison avec les formats de données propriétaires. Enfin, la notion d'espace de noms rend XML extensible et modulaire et offre la possibilité d'intégrer différentes structures XML. Ces derniers points ont été à l'origine de la création de nombreuses grammaires consacrées à la représentation de données biologiques. Citons dans ce sens, MAGE-ML[19] [SMS+02], BSML[20], SBML[21] [HFB+04], ou encore INSDseq[22] que nous allons détailler ci-dessous.

Tous ces langages ont pour objectifs de (i) définir le vocabulaire nécessaire afin de représenter de manière partagée l'information associée à une problématique biologique et de (ii) mettre en place des formats d'échange entre les applications biologiques.

MicroArray Gene Expression Markup Language (MAGE-ML) permet de décrire et de transférer des données d'expériences de transcriptomique (notamment à partir de puces de type MicroArray). Il propose dans ce cadre plusieurs sous-structures relatant de la conception des puces de leur fabrication , de l'expérience proprement dite ainsi que des données qui en résultent.

BSML donne une représentation des séquences biologiques et décrit en outre diverses représentations graphiques tel que des gels d'électrophorèses ou des alignements multiples.

Le Systems Biology Markup Language (SBML) permet de représenter les modèles généraux des réseaux de réactions biochimiques telles que les réseaux métaboliques, les voies de signalisation cellulaire, les réseaux de régulation, etc. SMBL propose une modélisation suffisamment générique qui justifie son exploitation dans plus d'une centaine de logiciels.

INSDSeq est une structure XML collaborative pour faciliter l'échange de données de séquences entre les membres du consortium des banques de séquences nucléotidiques (GeneBank, EMBL, DDBJ).

Comme nous venons de le voir, les centres de ressources ont pris l'habitude depuis de nombreuses années de proposer aux biologistes un accès à des banques de données biologiques plutôt qu'à des bases de données. Ils gèrent néanmoins, de manière interne, les données biologiques au travers de bases de données le plus souvent relationnelles. Les Systèmes de Gestion de Bases de Données Relationnels (SGBDR) offrent en effet les meilleurs compromis pour ce qui concerne le stockage de gros volumes de données et la rapidité d'accès aux données en environnements multi-concurrents. Ces qualités sont renforcées par la présence au sein des SGBDR du langage standard de requête structuré SQL qui fait office à la fois de langage de définition des données, de langage de manipulation de données et enfin de langage de contrôle de données. Les SGBD intègrent enfin des mécanismes pour garantir la sécurité des transactions et des systèmes ou encore des mécanismes d'optimisation des requêtes qui font toute leur force. Dans un premier temps, les logiciels commerciaux tel que DB2, Oracle ou encore Sybase ont été préférés par les centres de ressources généralistes, pour implanter les bases de données internes. Aujourd'hui les choses ont progressivement évolué, les sources de données spécialisées

[19] MicroArray and Gene Expression-Markup Language http ://www.mged.org/Workgroups/MAGE/mage-ml.html

[20] Bioinformatic Sequence Markup Language http ://www.bsml.org/

[21] System Biology Markup Language http ://sbml.org

[22] International Nucleotide Sequence Database Sequence http ://www.insdc.org/XMLStatus.html

```
<EMBL_Services
xsi:noNamespaceSchemaLocation=
"http://www.ebi.ac.uk/embl/schema/EMBL_Services_V1.1.xsd">
 <entry accession="CL520694" version="2" dataClass="GSS"
   taxonomicDivision="PLN" created="2004-04-04" lastUpdated="2006-05-26"
   releaseCreated="79" releaseLastUpdated="87">
  <description>
MUL5B09 Flanking Sequence Tag of Oryza sativa T-DNA insertion lines
Oryza sativa (japonica cultivar-group) genomic, genomic survey
sequence.</description>
  <keyword>GSS</keyword>
  <reference>
    <citation id="1" type="journal article" name="Plant J."
    volume="39" issue="3" first="450" last="464" year="2004">
       <title>
       High throughput T-DNA insertion mutagenesis in rice: a first step
       towards in silico reverse genetics.</title>
       <author>Sallaud C.</author>
       </citation>
    <citationLocation begin="1" end="257"/>
  </reference>
  <comment> Contact: Guiderdoni...</comment>
  <feature name="source">
    <organism>
       <scientificName>Oryza sativa (japonica cultivar-group)</scientificName>
       <taxId>39947</taxId>
       <lineage>
       <taxon>Eukaryota</taxon> <taxon>Viridiplantae</taxon>
       </lineage> </organism>
    <qualifier name="cultivar"><value>Nipponbare</value> </qualifier>
    <qualifier name="mol_type"><value>genomic DNA</value> </qualifier>
    <qualifier name="clone_lib">
       <value>
      Flanking Sequence Tag of Oryza sativa T-DNA insertion line</value>
    </qualifier>
    <locationElement type="range" accession="CL520694" version="1" >
       <basePosition type="simple">1</basePosition>
       <basePosition type="simple">257</basePosition>
     </locationElement>
    </location>
  </feature>
  <sequence type="genomic DNA" length="257" topology="linear" version="1">
       tccaccaaaatatatgtgcaaattcgatctaaacgtccacaaacaaaaaagaccaattcag...
  </sequence>
 </entry>
</EMBL_Services>
```

FIG. 2.3 – Représentation d'une séquence sous le format XML d'EMBL

sont très souvent maintenues à l'aide de SGBDR libres comme PostgreSQL ou MySQL et sont rendues disponibles aux usagers au travers d'interfaces Web. L'émergence du Web tout comme la dynamique des logiciels et des langages libres ont entraîné l'apparition de multiples bases de données spécialisées maintenues par des communautés de scientifiques ne disposant pas nécessairement de gros moyens.

En terme de modélisation, les données biologiques sont des données qui sont naturellement fortement agrégées. Pour exemple, un tissu est une collection de cellules, qui à leur tour sont une collection d'organites, qui à leur tour sont une collection de molécules, etc. La représentation de la complexité des données biologiques s'accommode donc fort mal des limites imposées par le modèle relationnel et notamment par la première forme normale dans le modèle relationnel qui va rendre nécessaire la décomposition de l'information dans de multiples relations. Dans ce sens, certains SGBDR, comme Oracle ou PostgreSQL proposent une surcouche objet, avec la possibilité par exemple de définir des types complexes, qui va permettre de mieux rendre compte de la réalité des objets biologiques.

Les systèmes de gestion de bases de données orientée objet (SGBDOO) sont des systèmes qui allient les concepts hérités du paradigme objet à l'instar des langages de programmation dit objets, et les qualités propres des systèmes de gestion de bases de données. Ils offrent l'avantage de manipuler puis de de traiter l'information de manière uniforme et donc de s'affranchir du problème bien connu dit de distorsion des langages (impedance mismatch) que l'on rencontre dès lorsqu'il s'agit d'imbriquer des ordres SQL dans un langage hôte de type procédural ou objet. Les concepts objets sont également particulièrement adaptés pour retranscrire la complexité et la dynamique des objets biologiques. Pour illustration, les liens d'héritage, d'agrégation ou encore de composition sont particulièrement bien pris en charge par la philosophie objet et vont permettre de relater de la diversité des objets biologiques et de leurs multiples interactions. Les liens d'héritage vont, additionnellement, permettre de manipuler les objets à différents niveaux de granularité et ainsi de mener des études sous des points de vue plus ou moins macroscopique ou microscopique selon les besoins. Par exemple, une protéine peut être modélisée au travers de ses différentes structures : structure primaire (séquence d'acides aminés), structure secondaire (hélices alpha, feuillets beta, coudes, etc), structure ternaire (assemblage des différentes chaînes peptidiques), structure quaternaire ou encore être modélisée au travers de ses domaines fonctionnels et de ses interactions avec d'autres objets biologiques (métabolisme). Les classes d'objets prennent non seulement en charge l'état des objets mais aussi leurs comportements au travers de diverses méthodes. C'est là encore un avantage indéniable à mettre à l'actif des SGBDOO, les objets biologiques sont très fortement dynamiques et sont de plus très largement analysés au travers de multiples méthodes.

Il existe toutefois peu de bases de données biologiques supportées par des SGBDOO. Nous pouvons citer, dans ce sens, le système de gestion de données basé objet ACeDB[23](pour *A Caenorhabditis. elegans* DataBase) développé en 1989 dans le cadre du programme de séquençage du nématode *Caenorhabditis elegans*. Il s'agissait en 1989 de construire un système de gestion de données d'inspiration objet, entièrement dédié à l'exploitation de données issues de projets de séquençage. Il était courant de penser alors que les SGBDR n'étaient pas adaptés à la gestion de données biologiques. La tendance s'est inversée depuis. AceDB n'implémente

[23]http ://www.acedb.org/

toutefois pas toutes les caractéristiques d'un SGBDOO, son modèle est certes basé objet mais les classes AceDB sont dépourvues de méthodes par exemple. Il demeure encore très populaire dans la communauté biologique, notamment dans le contexte des projets de séquençage pour plusieurs raisons essentielles. AceDB est distribué en open source, il fournit un schéma de données générique pour la représentation de données issues de projets de séquençage et enfin il offre des interfaces graphiques de restitution de données particulièrement soignées et appréciées par les biologistes. EyeDB[24] [VBV99] est un autre système de gestion de bases de données orienté objet développé spécifiquement pour les besoins du projet de séquençage du génome humain au sein du CEPH (Centre d'Etudes du Polymorphisme Humain) et particulièrement pour l'exploitation d'objets cartographiques (notamment cartes physiques et génétiques du génome). EyeDB, plus difficile d'accès, n'a pas reçu le même succès qu'AceDB de la part de la communauté biologique. Enfin, certains SGBDOO commerciaux comme O2, ont également supporté le développement de bases de données génomiques. Le recours à des SGBDOO, à l'exception d'AceDB, dans le cadre de projets de génomique est cependant resté confidentiel. La raison en est essentiellement la difficulté du passage à l'échelle, dès lorsqu'il s'agit de gérer de gros volumes de données.

2.1.3 L'open source et partage des schémas de bases de données

Les ressources bioinformatiques (sources de données biologiques comme outils d'analyse) sont, actuellement, disponibles de manière massive sur le Web. Des communautés scientifiques, qui se sont fédérées autour du développement d'outils génériques, sur la base du développement solidaire au bénéfice de tous, en sont à l'origine. La distribution libre d'applications est ainsi devenue monnaie courante dans le domaine de la biologie. Reprenons l'exemple d'AceDB, développé pour le ver *C. Elegans*, qui a été adapté pour de nombreuses autres espèces (Uk Cropnet, Graingenes, TropGeneDB, etc). Dans ce contexte, le logiciel ainsi que le schéma de la base sont distribués librement. Pour des bases de données implantées sous des systèmes relationnels commerciaux, seules les données, ou encore seuls le schéma et les données sont distribués. Pour Gramene[25], une source de données génétiques et génomiques pour les graminés, le schéma et les données sont ainsi rendues accessibles. De nombreuses sources de données, cependant, proposent librement le téléchargement de l'ensemble de leurs applicatifs (e.g. ENSEMBL[26]). La communauté bioinformatique s'est également organisée autour de développements collaboratifs et open source pour tout ce qui concerne les ressources de calcul. Sous l'impulsion de groupes de recherche tels que l'équipe à l'origine d'Ensembl [HBB+02], dirigée par L. Stein et J. Stajich, des interfaces de programmations (APIs) qui proposent différentes fonctionnalités réutilisables pour l'analyse d'objets biologiques, ont été développées dans les langages de programmation les plus en vogue. BioPerl [SBB+02] est ainsi une sorte de boite à outils générique qui couvre les principaux besoins par exemple pour la comparaison de séquences ou encore pour la recherche de domaines fonctionnels au sein de ces séquences (sur la base de la définition d'expressions régulières). Les développements de ces APIs sont réalisés de manière collaborative et sont coordonnés par la fondation open-bio qui organise régulièrement à cet effet des conférences à l'image de BOSC[27] (Bioinformatic open source conference). Parmi les

[24] http ://www.eyedb.org/
[25] http ://www.gramene.org/
[26] http ://www.ensembl.org/index.html
[27] http ://www.open-bio.org/wiki/BOSC

projets supportés par la fondation open-bio, nous trouvons des APIs de développements (BioPerl, BioJava, BioPython, BioRuby ou encore BioPHP) ou encore la suite logicielle EMBOSS (European Molecular Biology open source software).

Le projet de modèle générique de base de données (GMOD[28]) bénéficie de la même mouvance. L'objectif est en effet de partager au sein de la communauté biologique un schéma de données générique, des outils de visualisation et d'édition de génomes, des outils de recherche bibliographique ou encore des outils de gestion d'ontologies. Le projet est international et est financé par le NIH et l'USDA Agricultural Research Service, avec la participation de membres issus de plusieurs projets de base de données, incluant les projets WormBase, FlyBase, Mouse Genome Informatics, Gramene, Rat Genome Database, TAIR, EcoCyc, et Saccharomyces Genome Database. Le schéma de base de données de GMOD (CHADO) a été pensé pour pouvoir être directement opérationnel sous les deux principaux SGBD distribués sous des licences open sources, à savoir PostgreSQL et MySQL. L'objectif est donc véritablement de donner à la majorité des communautés de biologistes et/ou de bioinformaticiens de disposer de leurs propres ressources de travail en accompagnement des ressources déjà disponibles depuis le Web.

2.2 Les défis de l'intégration de données

Les efforts réalisés par la communauté biologique pour partager données et applications relèvent de ce que l'on pourrait appeler le patrimoine collectif. Par ailleurs, les enjeux scientifiques changent progressivement, il ne s'agit pas seulement aujourd'hui de produire de l'information expérimentale brute mais aussi de l'enrichir au travers d'analyses et d'expertises ou encore de l'enrichir à la lumière de données complémentaires. L'intégration de données est vue ici comme le processus nécessaire à l'assemblage de données, lequel va se révéler essentiel pour faire émerger de nouvelles données, de nouvelles expertises, de nouvelles hypothèses ou de nouveau modèles biologiques. A l'image des exemples présentés dans le chapitre 1.4, les scientifiques peuvent interpréter leurs résultats grâce à des données fournies par d'autres, les valoriser en y ajoutant l'information trouvée dans d'autres ressources distantes, voire également les confirmer en trouvant des résultats similaires. L'intégration peut être réalisée manuellement ou bien être automatisée à des degrés divers. Elle n'en reste pas moins un processus complexe qui va nécessiter un engagement fort de la part des communautés qui vont en avoir la charge.
Cette intégration pour qu'elle soit effective et fiable doit relever certains défis : prendre en compte la diversité des données biologiques, s'adapter face à l'autonomie des sources et s'affranchir de leur hétérogénéité tant sur le plan syntaxique que sémantique.

2.2.1 La diversité et autonomie des sources à intégrer

L'information biologique n'est pas centralisée dans une seule source mais distribuée dans de multiples sources de données. La centralisation faciliterait grandement l'intégration mais à contrario, en appauvrirait le contenu informationnel [Ste03]. En effet, la diversité des sources de données reflète l'expertise et la différence des points de vue des groupes qui les maintiennent. La centralisation peut entraîner des pertes d'informations. Par exemple les données ayant le même

[28]The Generic Model Organism Database http ://www.gmod.org

thème peuvent avoir un niveau de détail (granularité) différent si les compromis de stockage vont dans le sens de la performance, il y a alors appauvrissement du schéma de données. La meilleure solution pour conserver une grande latitude dans les données biologiques reste donc la diversité des sources, en gardant bien sûr possible, l'éventualité de leur unification [Ste03].

La majorité des sources fonctionnent de manière autonome, ce qui sous-entend, qu'elles sont libre de modifier leurs schémas ou leurs contenus sans le notifier publiquement. Cette autonomie peut être source de difficultés. Prenons l'exemple des liens supportées par les références croisées, un changement de nomenclature dans la source référencée peut ainsi entraîner une inconsistance de l'information dans la source référente [HK04]. Les sources de données peuvent également refuser temporairement leur accès pour cause de maintenance et ceci sans mettre en place d'alternatives. Cette instabilité dans la disponibilité des sources est renforcée par les limites du trafic sur le réseau qui contraignent leurs accès.

2.2.2 Hétérogénéité des sources de données

Le partage des données biologiques à travers de multiples sources mettent en lumière des différences de représentation. Ces sources qu'elles soient généralistes ou spécialisées n'ont pas forcément les mêmes objectifs et points de vue dans la représentation de l'information biologique. Par exemple, les données contenues dans la banque de structures protéiques PDB et la banque de séquences protéiques SWISS-PROT traitent de différents aspects des protéines. Par continuité, leurs formats d'échange tout comme leurs interfaces de restitution sont différents. L'hétérogénéité se situe donc à plusieurs niveaux. L'hétérogénéité du contenu des sources de données est justifiée puisque qu'elle rend compte de la richesse du patrimoine collectif et de la multiplicité des points de vue des biologistes. Dans cette section nous allons aborder les différents niveaux d'hétérogénéité, que cela soit au niveau de la structure ou du contenu.

2.2.2.1 Hétérogénéité syntaxique

Hétérogénéité dans les formats Il s'agit de l'hétérogénéité dans la manière dont est représentée l'information. On parle alors d'hétérogénéité du modèle de données. Les données génomiques sont, à ce titre, disponibles dans des formats non structurés (i.e. fichiers textes, ou HTML) également semi-structurés (XML) ou bien encore dans des formats dits structurés (réseau, relationnel, objet, etc) Au delà des formats, l'hétérogénéité perdure au sein de la conception elle même des modèles. Ainsi deux modèles relationnels décrivant les mêmes objets biologiques mais définis par des concepteurs différents vont posséder un certain nombre de différences, par exemple dans le nombre de relations, dans le nom attribué aux relations, dans le nom et le type de données décrivant les attributs ou encore dans les définitions de contraintes posées sur les relations et les attributs. Le travail de modélisation, bien que guidé par différentes méthodologies, reste une activité subjective.

Hétérogénéité dans les modes d'accès aux données Les protocoles d'accès aux données varient énormément par rapport aux sources. Par exemple, deux sources de données relationnelles peuvent proposer des méthodes d'accès différentes, une connexion anonyme à la base pour l'une et un téléchargement du fichier de "dump" pour l'autre. Très souvent les protocoles de communication utilisés sont HTTP, FTP ou encore RMI.

2.2.2.2 Hétérogénéité sémantique

Hétérogénéité de point de vue Chaque source contient des informations sur les mêmes objets biologiques sans qu'elles abordent toutefois le même thème ou point de vue. Par exemple le thème de SWISS-PROT est la séquence protéique, celui de PDB est la structure 3D de la protéine.

Définition divergente Les sources de données peuvent différer dans leurs manières de représenter les concepts clés autour des entités décrites [ELR01]. Par exemple, GenBank définit un gène comme une annotation sur une séquence (la séquence est stable mais le gène peut changer, être renommé) alors que MGD[29] représente un gène comme un locus qui confère un phénotype (la notion de gène est cette fois une notion stable).

Hétérogénéité des valeurs Une diversité intrinsèque peut être trouvée dans les sources différentes malgré la correspondance des concepts. Il s'agit des valeurs qu'ils peuvent prendre. Par exemple, une longueur de tige aura une valeur de 10 centimètres dans une source et une valeur de 100 millimètres dans une autre, elle pourra également avoir la valeur 4 dans un troisième qui stockera des valeurs correspondant à des échelles de tailles.

Hétérogénéité de nomenclature Pour les sources possédant une structure (structurée ou semi-structuré) il existe une hétérogénéité au niveau des éléments et leurs relations. En effet les noms qui sont donnés à ces éléments et relations peuvent être différents selon les schémas entraînant des conflits sémantiques. Un même nom attribué à des concepts différents (le concept plante peut signifier plante sauvage et plante OGM) serait une homonymie alors qu'un nom différent attribué à des concepts identiques serait une synonymie.

2.3 Standardisation des données

Intégrer des données provenant de sources variées représente un besoin essentiel pour les biologistes. Pour que les informations puissent être comparables il faut cependant que les concepts soient identiques. Or les conflits tant syntaxiques que sémantiques sont nombreux entre les sources qui sont partagées sur Internet. Ontologies et méta-données permettent de réduire ces conflits. La conception d'ontologies permet de formaliser et de structurer les concepts d'un domaine d'intérêt. Les méta-données sont des standards de descriptions pour les sources de données publiées sur internet.

2.3.1 Les méta-données

Actuellement, l'information disponible sur Internet est suffisamment compréhensible pour une personne mais reste insuffisante pour que les machines puissent l'exploiter et l'interpréter correctement (i.e. sans intervention humaine). Le langage HTML, encore majoritairement présent sur Internet, est en effet inapproprié pour ce type de traitement. Il n'est pas clairement structuré (par exemple, il ne sépare les données ni de leur structure ni de leur mise en forme) et possède une variabilité importante dans la manière de représenter un document ce qui rend difficile son analyse pour produire une indexation cohérente. La solution préconisée par le consortium W3C est de le remplacer par des langages plus adaptés. De plus, afin de permettre une indexation et une recherche facilitée, la volonté est alors de décrire les documents Web

[29] Mouse Genome Database

avec des données additionnelles décrivant leurs contenus ou encore méta-données. Les méta-données sont dans, un premier temps, des données à part entière et dans un second temps des " données sur des données "[30]. A l'inverse des données "classiques", elles possèdent un niveau d'abstraction (le niveau méta) qui leur permettent de venir en support d'autres données. Le dictionnaire de données, classiquement trouvé en support des bases de données dans les SGBDR, relève typiquement de cette définition de méta-données. Au niveau du Web, les méta-données seront présentes au niveau de structures (documents bien souvent décrits en RDF) additionnelles qui vont venir annoter les documents Web, participant donc à rendre les ressources auxquelles elles sont associées compréhensibles et exploitables par les machines. Pour plus d'efficacité, les méta-données sont normalisées. La norme Dublin Core a été mise en place, à cet effet pour le Web et par le W3C, depuis 1995. Dublin Core définit quinze éléments dont la sémantique a été établie par un consensus international de professionnels provenant de diverses disciplines. Ces éléments sont répartis autour de trois domaines, qui permettent d'identifier et de décrire les ressources du Web : (i) contenu (titre, sujet, description, source, langage, relation, couverture) , (ii) propriété intellectuelle (créateur, éditeur, contributeur, droits) , (iii) matérialisation (date, type, format, identifiant).

Dans le domaine biomédical, Markowitz et al. ont identifié les méta-données suivantes comme étant nécessaires pour décrire des bases de données [MCKS97] :

1. des informations générales incluant le nom de la source, son adresse (URL), le langage dans lequel la source est décrite, la manière dont elle est implémentée et des mots-clés permettant des recherches de haut niveau dans la source ;

2. le schéma définissant la structure de la source, voire des définitions associées aux différents éléments du schéma ;

3. des vues représentant des interprétations alternatives des sources en fonction des utilisateurs ;

4. les références croisées connues qui référencent d'autres sources.

Parmi les langages du Web sémantique, le RDF (Resource Description Framework) est un moyen de décrire, échanger et réutiliser des méta-données structurées. Il s'agit d'un langage qui étend XML développé par le W3C et qui a fait l'objet d'une "Recommandation" en 1999. Toutefois, RDF ne précise pas la sémantique des ressources décrites par les différentes communautés d'utilisateurs de méta-données.

Les méta-données participent à l'interopérabilité des sources en assurant l'échange et le partage d'informations rendues lisibles et exploitables par les machines. Grâce à l'utilisation de standards elles permettent d'homogénéiser la description des sources réparties sur Internet, et ce de façon tout à fait indépendante des détails de représentation propres au contenu de chaque source. Cependant, en pratique elles ne sont que très rarement ou très partiellement fournies par les auteurs des sources ; et quand elles sont fournies, elles le sont parfois dans un format difficilement exploitable. De plus, homogénéiser les formats de méta-données tout comme maintenir ces dernières à jour peut se révéler un travail complexe et fastidieux.

[30] http ://www.w3.org/DesignIssues/Metadata.html

2.3.2 Les ontologies

Le terme ontologie[31] a été introduit par ARISTOTE qui l'a défini comme étant la science de l'Être. Le domaine de l'IA (Intelligence Artificielle) a repris, de manière beaucoup plus récente, la notion d'ontologie à son compte. Gruber [Gru93] définit ainsi une ontologie comme **la spécification explicite d'une conceptualisation**. Plus simplement, une ontologie représente l'ensemble des connaissances d'un domaine, au travers d'une hiérarchie de concepts liés par des relations sémantiques. Les notions de partage et de réutilisabilité sont des notions importantes qui caractérisent les ontologies. Une ontologie va en effet être construite pour les besoins d'une communauté et va permettre d'expliciter les connaissances qui sont partagées par l'ensemble de la communauté. Il est donc impératif qu'elle reçoive l'adhésion de tous. Par continuité, les ontologies peuvent être réutilisées par d'autres communautés. Les mécanismes de raisonnement qui peuvent s'appliquer sur les ontologies, s'avèrent également des caractéristiques essentielles. Différents modes de raisonnement vont alors permettre d'évaluer la qualité de l'ontologie construite (classification, satisfiabilité, etc) ou encore faire émerger de nouvelles connaissances (déduction par exemple). A la différence des bases de données objets ou relationnelles, les ontologies font l'hypothèse du monde ouvert laissant place à l'incomplétude et à l'évolutivité de la connaissance. Des langages comme des langages à base de frames (ou à base de prototypes) et des logiques de description permettent de manipuler des données incomplètes (i.e. slots absents) ou des données non figées (i.e. reclassement de concepts ou de rôles). Des mécanismes chez les langages à base de frames permettent également de tenir compte du bruit (i.e. des erreurs) sur un jeu de données.

Le développement du Web Sémantique a fait sortir le concept d'ontologie du domaine confidentiel de l'I.A. Selon Tim Berners-Lee [BLHL01], l'expression Web Sémantique fait référence à la vision d'un Web où il n'y aurait pas de distinction entre humains et machines que cela soit pour l'échange ou pour l'exploitation de ressources. Les ontologies occupent, dans ce contexte, une place centrale et vont jouer un rôle clé (i) dans la structuration et l'exploitation des métadonnées (cf. section 5.1.2.1), (ii) comme représentation pivot pour l'intégration de sources de données hétérogènes (cf. section 3.1), (iii) dans la description des services Web et, en général , partout où il va être nécessaire d'appuyer des modules logiciels sur des représentations sémantiques nécessitant un consensus.

2.3.2.1 Représentation d'une ontologie

Une ontologie peut se représenter généralement au travers de langages issus de l'IA (logiques de descriptions, langages à base de frames, graphes conceptuels, etc) ou encore au travers de langages qui ont émergé avec le Web Sémantique (OIL, OWL, etc) qui sont souvent inspirés des logiques de description et qui se basent sur XML pour leur syntaxe. Nous faisons une introduction aux langages qui nous semblent les plus représentatifs en terme de construction d'ontologies.

Les langages à base de frames Les langages à base de frame , Minsky [Min75] s'appuient sur les notions de prototype (ou frame ou encore schéma), d'objet, de classe et d'instance pour représenter la connaissance. A la différence des langages objet, ces langages prennent en charge l'incomplétude et l'évolutivité des connaissances d'ou le concept de prototype. Les frames sont caractérisés par un ensemble de slots (attributs). Les slots sont

[31] Définition que l'on retrouve dans le Petit Robert : *la partie de la métaphysique qui s'intéresse à l'Être en tant qu'Être*.

ensuite typés au travers de facettes : slot monovalué, multi-valué, type primitif ou complexe (pointant sur un autre frame), restriction au travers d'un domaine de valeurs, valeur par défaut, etc. Les langages à base de frames proposent des primitives de modélisation très riches et sont, de ce fait, pénalisés pour tout ce qui ce qui concerne les mécanismes de raisonnement automatique par rapport à d'autres langages comme les logiques de descriptions.

Les logiques de descriptions Elles sont issues d'un courant de recherche initié par le système KL-ONE [BS85]. Les logiques de description héritent des expériences faites sur des langages logiques tels que les logiques de prédicats mais également sur des approches réseaux telles que les réseaux sémantiques et les langages de frames. Elles sont encore appelées logiques terminologiques ou en anglais description logics (DLs).

Le formalisme des logiques de descriptions utilise comme notions de base (i) le concept, qui représente une entité générique regroupant un ensemble d'individus partageant les mêmes propriétés, (ii) le rôle, qui est une relation binaire entre les concepts et (iii) l' individu, qui sera une instance du concept qui le représente [BCM+03].
Ces éléments sont organisés en deux niveaux. Le niveau terminologique ou **Tbox** est bâti à l'aide de concepts et de rôles (i.e. les relations binaires) ainsi que d'opérateurs spécifiques à la logique permettant par exemple de limiter le nombre d'instances d'un même rôle (e.g. un gène à au moins un transcript). Le niveau assertionnel ou **Abox** fournit des instances des concepts et des rôles.

Concepts et rôles se construisent à l'aide d'un ensemble de constructeurs et de propriétés. Un concept ayant des propriétés atomiques ou incomplètes sera un concept dit **primitif**. Un **concept défini** aura des propriétés complètes construites à partir de concepts primitifs ou définis et à partir de constructeurs additionnels. Les rôles se distinguent également entre **rôles primitifs et rôles définis**. Les rôles peuvent être précisés dans leur relation avec le concept au travers de restrictions. Par exemple une restriction sur la cardinalité d'un rôle, fixe le nombre minimal et maximal d'instances de concept qui peuvent être attachées à ce rôle.

La **subsomption** est la relation qui permet de structurer les concepts et les rôles en hiérarchies. On dit qu'un concept A subsume un concept B si A est plus général que B ce qui se peut se traduire par : l'ensemble des individus de A contient l'ensemble des individus de B. Ce type de relation est appelé relation de spécialisation et peut être étiqueté "est-un" (is-a en anglais).

Dans les logiques, on trouve deux opérations basées sur la subsomption qui sont à la base du raisonnement terminologique.

La **classification** est une opération qui permet d'insérer des nouveaux concepts ou rôles dans la hiérarchie. Elle ne s'applique toutefois que pour les concepts et rôles définis

L'**instanciation** permet de retrouver des concepts rattachés à un individu.

Les caractéristiques d'une logique se mesurent par son pouvoir expressif, ses mécanismes de classification et de vérification de la cohérence de concepts. Un langage très expressif possédera de nombreux constructeurs. En revanche une forte expressivité dessert souvent la performance des mécanismes de classification de concepts. Tous ces paramètres sont pris en compte lors de l'utilisation d'un langage dans une application.

Les ontologies publiées sur le Web s'appuient sur des standards tels que DAML+OIL [Hor02] ou plus récemment OWL [32] [DS04].

DAML+OIL est un langage qui emprunte ses primitives de modélisation aux logiques de descriptions, sans en reprendre toutefois complètement la terminologie. Il va décrire un domaine d'intérêt en terme de classes et de propriétés. Sa richesse en terme de constructeurs disponibles le rend très expressif. Il s'appuie syntaxiquement sur les langages standardisés du Web tels que XML (eXtended Markup Language), RDF (Ressource Description Framework) et RDF Schéma.

OWL est le successeur de DAML+OIL. Il mélange l'expressivité des logiques et les propriétés des frames. Il est décliné en trois versions, OWL-lite, OWL-DL et OWL-full, variant en fonction de l'expressivité et de la complexité du langage (complétude) . OWL-full a un fort pouvoir expressif mais à l'inverse, il ne possède pas de mécanisme de décidabilité. OWL-DL est un intermédiaire. OWL-lite est le langage volontairement le moins expressif des trois. OWL Lite est inclus dans OWL DL et OWL DL est inclus dans OWL Full.

2.3.2.2 Alignement d'ontologies

La représentation d'un domaine nécessite soit la création *de novo* d'une ontologie soit la réutilisation de plusieurs ontologies (on parlera alors de fusion ou de recherche de correspondances). De manière générale, ce mécanisme de recherche de correspondances est retrouvé dans les approches d'intégration de données, et se nomme alors la correspondance de schémas (ou *schema mapping*). Le principe peut en être décrit brièvement, il s'agit de prendre en entrée deux schémas et de fournir en sortie une correspondance, c'est à dire un ensemble de relations sémantiques entre les entités (concepts, relations, instances) de ces schémas. Cette étape est soit, comme très souvent, réalisée manuellement ou encore au travers de méthodes automatisées. Par exemple, l'algorithme PROMPT [NM00], dont une implémentation est proposée dans le cadre du logiciel PROTEGE, permet (i) de gérer plusieurs versions d'une ontologie, (ii) de fusionner deux ontologies et (iii) d'extraire une partie d'une ontologie pour la replacer dans une autre.

2.3.2.3 Des éditeurs d'ontologies

De nombreux éditeurs permettent aujourd'hui de construire des ontologies. Il n'existe pas de monopole d'utilisation. Le choix d'un logiciel dépendra (i) de son interface conviviale et de sa prise en main, (ii) du modèle utilisé pour construire les ontologies : basé le plus souvent sur des logiques de description ou des langages à base de frames, (iii) de la présence d'un moteur d'inférence, (iv) de l'évolutivité du logiciel qui se traduit par la réutilisation du code pour l'ajout de fonctionnalités nouvelles ou de "plugins" (modules d'ajout). Nous illustrons ce paragraphe par trois exemples d'outils reflétant de leur diversité : PROTEGE, OilEd et Dag-edit.

PROTEGE[33] **[NSD+01]** est un environnement graphique de conception et de gestion d'ontologies développé par le Standford Medical Informatics. L'outil est gratuit, développé en Java et s'installe localement. PROTEGE fonctionne selon un modèle à base de frames [NFM00]. Récemment, la possibilité de développer des ontologies en OWL a été ajoutée grâce à un plugin adapté [KFNM04]. La force de PROTEGE repose sur sa modularité et son extensibilité et sur une communauté dynamique qui l'exploite régulièrement et qui le

[32] Web Ontology Language

fait évoluer en apportant sans cesse des améliorations. En conséquence, PROTEGE met à la disposition de tous un nombre considérable de plugins apportant différentes fonctionnalités (outils de visualisation graphique des ontologies, interfaces de consultation, outils de conversion, modules de connexion avec des bases de données relationnelles etc). Son ouverture à de nombreuses communautés, augmente les possibilités de transferts technologiques et rend disponible en outre plusieurs dizaines d'ontologies mises gracieusement à disposition par leurs auteurs.

OilEd[34] **[BHGS01]** est également un outil graphique de création et d'édition d'ontologies supportées par le langage OIL [FHvH+00] développé à l'université de Manchester. L'outil est gratuit et s'installe localement. Le modèle de connaissances est basé sur les logiques de descriptions. En contraste des systèmes à base de frames, OilEd permet une meilleure définition des restrictions. Cet éditeur utilise également les services d'un raisonneur, FaCT [Hor99], qui permet de tester la satisfiabilité des définitions de classes et de découvrir des subsomptions restées implicites dans l'ontologie.

Dag-Edit[35] est un outil graphique permettant de naviguer, rechercher et éditer Gene Ontology ou les ontologies basées sur le modèle de graphe acyclique orienté (Direct Acyclic Graph). Cet outil est très orienté bioinformatique car il a été développé dans le cadre du projet Gene Ontology. Il permet de construire une ontologie en se basant sur deux types de relations, à savoir, *is-a* et *part-of*.

2.3.3 Les ontologies et les méta-données dans le domaine biologique

2.3.3.1 Gene Ontology

Gene Ontology ou GO[36]est un vocabulaire contrôlé et structuré qui fut développé initialement dans le but d'annoter les produits de gènes dans les organismes eucaryotes [ABB+00, Con01, HCI+04]. Le projet GO débuta en 1998 par une collaboration entre trois bases de données d'organismes modèles : FlyBase (Drosophile), the Saccharomyces Genome Database (SGD) and the Mouse Genome Database (MGD). Ce Gene Ontology Consortium, est en charge de faire évoluer GO. Il est composé aujourd'hui de scientifiques issus des plus importantes banques de données génomiques actuelles dans les domaines des plantes, animaux et organismes microbiens. GO est organisée en 3 sous-vocabulaires : les processus biologiques, les fonctions moléculaires ainsi que les composants cellulaires.

(i) Les fonctions moléculaires, (molecular_function), décrivent les activités individuelles des produits de gènes (ex : carbohydrate binding, ATPase activity, etc.). (ii) Les processus biologiques (biological_process), décrivent à un niveau général les grandes fonctions de l'organisme (ex : la mitose ou métabolisme des purines). (iii) Les composants cellulaires (molecular_component), décrivent la localisation cellulaire du produit de gène, la structure cellulaire du composant ou encore les complexes macromoléculaires.

Ces vocabulaires sont structurés sous la forme de graphes acycliques orientés ou DAG (Directed Acyclic Graph). Cette représentation constitue un réseau de nœuds et d'arêtes dans laquelle les premiers représentent un terme GO (e.g. 'development', GO :0007275) et les deuxièmes les relations (e.g. 'is_a', 'part_of'). Chaque terme est relié par la relation de spécialisation/généralisation 'is_a' ou la relation de composition 'part_of' ce qui confère à l'ontologie GO une structure hiérarchique. Le nombre de relations entre les termes n'est pas limité : chaque terme enfant peut

[36]http ://www.geneontology.org/

avoir plusieurs termes parents.

Une des contraintes fondamentales dans l'évolution de GO est le respect de la règle du 'True Path Rule' qui signifie que les propriétés d'un terme enfant doivent pouvoir s'appliquer aux termes parents. Ce qui va complètement à l'opposé de la logique de programmation objet par exemple. En général la spécialisation permet de faire hériter les propriétés des termes parents vers les termes enfants. La mise en oeuvre de cette règle oblige une spécialisation des termes en fonction du contexte (e.g. 'embryonic development (sensu Metazoa) GO 0009792' spécialise 'embryonic development GO :0009790'). Malgré la mise en place de règles, le maintien de la cohérence est difficile face à l'évolution sans cesse constante de l'ontologie qui aujourd'hui dépasse les 18000 termes. L'utilisation de langages formels peut lever des ambiguïtés ou des incohérences, ce qui a été démontré dans (Yeh et al [YKNA03]).

Le rôle de GO dans l'annotation des données génomiques et des produits de genes est aujourd'hui incontestable (un millier d'articles en fait référence dans Pubmed). Le projet GOA[37] (Gene Ontology Annotation) réalise le lien entre les termes GO et les protéines décrites dans UniProt [CBL+04, CMB+03, CMB+04].
Les termes associés aux produits de gènes ont des codes assignés correspondant au type d'annotation réalisé (Table 2.1). Les codes influent directement sur la qualité de l'annotation (e.g. une IC annotation manuelle aura plus de poids sur la valeur de l'annotation).

Code	Origine de l'annotation
IC	Inferred by Curator
IDA	Inferred from Direct Assay
IEA	Inferred from Electronic Annotation
IEP	Inferred from Expression Pattern
IGI	Inferred from Genetic Interaction
IMP	Inferred from Mutant Phenotype
IPI	Inferred from Physical Interaction
ISS	Inferred from Sequence or Structural Similarity
NAS	Non-traceable Author Statement
ND	No biological Data available
RCA	Inferred from Reviewed Computational Analysis
TAS	Traceable Author Statement
NR	Not Recorded

TAB. 2.1 – Types d'annotation des termes GO

Toutes les données et les outils de manipulation d'ontologies GO sont disponibles par téléchargement à partir des sites de GO et GOA. Certains outils tels que des GO Browsers (e.g. Amigo) et des éditeurs (e.g. Dag-Edit) sont très utilisés.

Amigo [38] est une interface Web de visualisation des ontologies GO. Il permet la navigation dans la hiérarchie de termes de manière graphique et interactive. Un formulaire permet de faire une recherche de termes ou de noms de gènes avec la possibilité de restreindre par nom d'espèces, sources de données ou codes d'annotation.

DAG-Edit est un éditeur d'ontologie GO proposé par le Consortium [Con01] permettant de modifier GO et de faciliter sa navigation.

[37] http ://www.ebi.ac.uk/GOA/

Le développement important d'ontologies formalisant les nombreux domaines de la biologie a conduit à la création d'un portail : OBO[39]. Il compte aujourd'hui près de 60 ontologies. La plupart d'entre-elles sont construites suivant le même modèle de données que GO (e.g. format OBO). Certaines sont génériques, alors que d'autres sont spécifiques d'un organisme (e.g. Arabidopsis gross anatomy, Maize gross anatomy).

Afin que ces ontologies puissent être réutilisables et en outre puissent être fusionnées, leur langage de représentation devait inclure des mécanismes de raisonnement et de classification. Une évolution de GO vers le langage DAML+OIL puis OWL [WSGA03] est à la base d'un projet *Gene Ontology Next Generation* (GONG)[40] qui a pour but de fournir des outils et méthodes pour la migration d'ontologies.

2.3.3.2 EcoCyc

EcoCyc[41] [KRS+00] est une base de connaissances développée pour le modèle biologique Escherichia coli K12. L'objectif d'EcoCyc est d'intégrer des données biologiques ainsi que des annotations sur les voies métaboliques, les transporteurs et les réseaux de régulations. Cette base est consultable à partir d'une interface Web ou d'un outil de visualisation de voies métaboliques téléchargeables.
EcoCyc a développé une ontologie pour décrire la complexité du domaine considéré et construire un schéma de base de connaissances. Les classes de cette ontologie composent le schéma alors que les instances de ces classes, composées d'attributs et de valeurs, composent les faits. EcoCyc utilise le langage à base de frames Ocelot, dont les capacités sont similaires de celle d'HyperTHEO [KP96], pour décrire son ontologie.
D'après les auteurs, l'expressivité des frames permet de capturer toute la complexité contenue dans l'information biologique et de s'adapter facilement aux évolutions fréquentes des schémas des bases de données biologiques qui sont exploitées en complément dans EcoCyc. De plus, HyperTHEO possède des capacités d'inférences basées sur des règles, un langage de contraintes pour maintenir l'intégrité de donnée et un langage de requêtes. Le noyau de la base de connaissance décrit le génome, le métabolisme ainsi qu'une taxonomie de composés chimiques qui permet la description des classes ADN, ARN, polypeptides et protéines. Dans l'ontologie, les chromosomes sont faits d'ADN, les gènes sont des fragments d'ADN et sont positionnés sur les chromosomes. Les voies métaboliques sont des collections de réactions qui agissent sur les composés chimiques. L'avantage d'utiliser une ontologie pour décrire le schéma porte sur son expressivité ainsi que sa réactivité face aux évolutions qui affectent régulièrement l'information biologique. La manipulation de l'ontologie est transparente pour l'utilisateur. Elle est utilisée comme schéma générique au niveau des bases qui sont développées dans ce projet par exemple dans MetaCyc qui est une extension dédiée aux voix métaboliques multi-espèces [CFF+06, KOMK+05, ZFT+05, KPKZ04, KZM+04, KRPPT02, KRS+00].

2.3.3.3 TAMBIS

« Transparent Access to Multiple Biological Information Sources » (TAMBIS) est un projet de recherche visant à unifier l'accès à différentes sources de données, à l'usage de la communauté biologique. TAMBIS s'appuie à cet effet sur une architecture de médiation et notamment

[39] Open Biomedical Ontologies, http ://obo.sourceforge.net/
[40] http ://gong.man.ac.uk/downloads/
[41] http ://EcoCyc.org

sur une ontologie nommée TaO (Tambis Ontology) qui va supporter les activités de médiation. L'architecture de TAMBIS est basée sur un modèle classique de médiateur/adaptateur à trois niveaux [Wie92].

Le premier niveau se compose d'une ontologie ainsi que d'une interface de navigation utilisateur qui exploite l'ontologie. A l'aide de l'interface, l'utilisateur forme une requête en combinant les termes de l'ontologie.

Le deuxième niveau est un niveau de médiation qui identifie les sources appropriées pour satisfaire la requête et la réécrire en une série d'opérations sur les sources identifiées.

Le troisième niveau comprend une série de sources ainsi que des adaptateurs.

En ce qui concerne TAMBIS ontologie (Tao), la motivation est de capturer les connaissances biologiques et bioinformatiques dans un langage permettant aux concepts et à leurs relations d'être interprétées par les ordinateurs.

Tao utilise la logique de description GRAIL [RBG+97, RN94, RRZ+03] comme langage de représentation des connaissances. GRAIL était originellement développée pour modéliser une terminologie médicale dans le but de fournir un support pour une interface clinique utilisateur. Ce premier niveau fait la distinction entre le domaine des concepts et le domaine des rôles. Dans ce premier domaine sont décrits des concepts généraux traitant des structures, substances, processus et fonctions. Dans le domaine des rôles sont définis des relations génériques telles que la collection et la localisation. Cette hiérarchie de haut niveau a été étendue dans TAMBIS avec une hiérarchie de bas niveau représentant les connaissances utilisateurs dans le domaine biologique. Ce modèle est centré sur la description des concepts Protéines, Acides Nucléiques et les concepts qui en découlent comme Enzyme, DNA et RNA . Les fonctions et processus biologiques sont aussi décrits. Tous ces concepts sont présents dans une hiérarchie de type " is a kind of " mais d'autres relations comme "is component of " enrichissent le modèle grâce aux rôles. 1800 concepts sont décrits traitant des séquences protéiques de SwissProt, des motifs et structures de Prosite, des enzymes et voies métaboliques de Enzyme DB, des EST de DB EST. Des concepts sont aussi construits sur les homologies de séquence et sur la taxonomie définie selon le NCBI. Dans le projet TAMBIS, l'ontologie sert de support pour la formulation des requêtes utilisateurs. En effet, les requêtes sont exprimées à partir des termes plutôt que directement sur les sources. L'ontologie joue un rôle de médiateur entre les sources, réglant les conflits sémantiques au niveau de l'abstraction.

Chapitre 3

État de l'art sur l'intégration

Sommaire

DEPUIS plusieurs années de nombreux systèmes d'information biologique sont développés en utilisant et adaptant les technologies de l'informatique les plus pointues. Le nombre important de systèmes reflètent l'intérêt et l'expertise des groupes qui les maintiennent. A l'image des données et des connaissances, ils sont en perpétuelle évolution. Leur rapprochement est souvent réalisé à partir de compromis divers qui peuvent malheureusement appauvrir les ressources finalement mises à disposition [Ste03].

L'approche intégration a pour objectif de faciliter l'accès de manière "uniforme" à des sources de données multiples, réparties et hétérogènes à travers une interface web. Ceci permet de pallier les problèmes de l'interopérabilité des sources [Kar95] c'est-à-dire au manque de standard de représentation des données et à la communication inter-application. Il existe différentes approches pour aborder l'intégration de manière générale. Les systèmes d'intégration peuvent être globalement catégorisés en deux approches : matérialisée et virtuelle ([HK04]). Dans l'approche matérialisée, les ressources sont centralisées dans un même système (dupliquées localement) ce qui permet une grande maîtrise de l'intégration et une sécurité au niveau de la confidentialité des données. A l'inverse, une approche virtuelle ne stocke pas les ressources localement mais centralise leurs schémas. Elle a plus de souplesse dans l'ajout de nouvelles sources, mais est pénalisée sur les traitements complexes. Dans ce chapitre, nous allons présenter rapidement les caractéristiques qui sous-tendent la notion d'intégration et permettront ensuite d'évaluer les systèmes la mettant en oeuvre (les caractéristiques devenant alors des critères d'évaluation). Ensuite, nous décrirons l'approche matérialisée. Nous présenterons dans un premier temps l'architecture entrepôt de données, puis nous donnerons des exemples d'entrepôts en bioinformatique. Enfin nous traiterons de l'approche virtuelle et nous détaillerons notamment l'approche dite "navigationnelle" en donnant des exemples d'application. Puis nous décrirons la médiation en donnant également des exemples d'application dans ce domaine.

3.1 Critères d'évaluation des approches d'intégration

Afin de distinguer précisément les différentes approches d'intégration, plusieurs critères peuvent être évalués : le **type de données** qu'elles intègrent, le **type d'intégration**, le **modèle pivot** qu'elles utilisent, le **degré d'intégration sémantique**, le **niveau de transparence** fourni à l'utilisateur, la construction du **schéma global** et le choix de la **localisation des sources**. Tous ces critères ne sont pas indépendants, certains sont la conséquence d'un choix précédent. Par exemple, un type d'intégration serrée implique la construction d'un schéma global. Dans les sections suivantes nous allons définir les différents critères permettant d'évaluer les approches d'intégration.

3.1.1 Formats des données intégrées

Le format de donnée correspond à la structure dans laquelle sont représentées les données. Un fichier Excel, une base de données ou un document XML sont tous les trois des formats de données différents. Selon leurs fonctionnalités, les systèmes d'intégration peuvent intégrer les données stockées sous des formats structurés, semi-structurés ou non-structurés.

Des données structurées, par exemple une base de données relationnelle, ont un schéma prédéfini, où chaque item est défini à partir de l'élément du schéma qui lui correspond. Des données semi-structurées ont une structure mais qui n'est pas définie sous la forme d'un schéma [BDH+95]. Un document XML est considéré comme un format de données semi-structurées. Chaque item contient sa propre sémantique généralement sous la forme d'un label. Toutefois, la somme de tous les labels des données semi-structurées peut être considérée comme son schéma. Les données non-structurées n'ont aucune structure, comme des documents textuels, des images ou des tableaux de données.

Dans les approches intégration, nous trouvons donc des systèmes distribués homogènes dans lesquels les données obéissent à un format commun, des systèmes distribués hétérogènes dans lesquels les formats sont variés.

3.1.2 Le type d'intégration

Dans les systèmes d'intégration, on distingue l'intégration à couplage fort (tight) pour laquelle les données des sources sont intégrées dans un schéma global, de celle dite à couplage lâche (léger ou loose) qui ne fournit pas de schéma, mais uniquement un langage pour interroger le contenu des sources de données. Ainsi, l'intégration à couplage fort fournit un schéma, un langage et une transparence d'interface alors que l'intégration lâche n'offre que la transparence [Gué05].

L'intégration à couplage fort

L'intégration à couplage fort fournit un schéma unifié (intégré ou global) comme interface du système. Ce schéma peut être créé selon un processus (semi-)automatique ou peut être créé manuellement. Il peut couvrir l'ensemble des données des sources ou uniquement une partie, mais doit conserver la sémantique des sources de données pour ensuite permettre la pertinence des requêtes. Pour assurer l'équivalence sémantique avec les sources de données et le système d'intégration, il faut établir des correspondances entre le schéma global et les schémas des sources. Ces correspondances peuvent être exprimées par des ontologies ou des définitions de règles d'appariement (cf. chapitre 2). Elles peuvent être exprimées à l'aide de langages ou bien inférées de manière automatique. Ce type d'intégration a l'avantage d'éviter à l'utilisateur final de devoir connaître tous les schémas des sources de données, mais plutôt d'avoir une connaissance unique du schéma global. D'un autre côté, il faut définir les correspondances entre les schémas des sources et le schéma global, ceci nécessitant l'implication d'experts du domaine [Gué05].

L'intégration lâche

L'intégration lâche ne fournit pas de schéma global pour l'interrogation du système, mais un langage de requête uniforme qui masque ainsi l'hétérogénéité des sources de données. C'est

alors à l'utilisateur de gérer cette hétérogénéité lors de ses requêtes. Pour faciliter l'accès aux données, ce type de système fournit généralement des vues intégrées. Les utilisateurs peuvent en effet définir des vues sur certaines données qui peuvent ensuite être accessibles pour des requêtes. Bien sûr, la frontière entre les deux types d'intégration reste difficile à cerner, le principal critère pour discerner les deux approches, reste celui de la visibilité (ou non) pour les utilisateurs des schémas des sources. Si dans l'intégration à couplage fort, ils ne sont jamais visibles, ils sont au contraire toujours visibles dans l'intégration à couplage lâche [Gué05].

3.1.3 Le modèle de données ou le modèle pivot

Les systèmes d'intégration repose sur un modèle de données pivot. Le modèle est celui dans lequel est exprimé le schéma global dans le cas d'une intégration à couplage fort, et il se base sur le langage de requête utilisé pour accéder aux sources dans le cas d'une intégration à couplage lâche. Le choix de ce modèle pivot reste délicat car il entraîne obligatoirement des transformations entre modèle des sources et modèle pivot. Ces transformations peuvent avoir diverses incidences et anomalies. Par exemple des incompatibilités surviennent si des données semi-structurées sont intégrées dans un système de données structurées. De même, des problèmes surviennent si des données provenant d'un modèle hautement sémantique doivent être intégrées dans un modèle plus pauvre. Par exemple, intégrer un schéma orienté objet dans un schéma relationnel induit une perte de connaissance, dans le sens inverse, ceci conduit à un enrichissement sémantique.

3.1.4 Les degrés d'intégration sémantique

Certains systèmes intègrent des sources de données complémentaires ne présentant pas d'objets équivalents et exportent donc certaines parties des schémas de celles-ci. D'autres systèmes, au contraire, intègrent des sources de données ayant des contenus chevauchant. Une agrégation d'information est alors requise pour identifier des objets équivalents d'un point de vue sémantique, c'est-à-dire décrivant le même concept. L'intégration d'informations complémentaires est appelée « intégration horizontale » tandis que l'intégration de données chevauchantes est appelée « intégration verticale » [Suj01]. Dans le cas d'une intégration verticale, on distingue différents niveaux d'intégration sémantique selon que les données sont collectées sans aucune recherche d'équivalence parmi les objets issus des différentes sources, ou fusionnées afin d'identifier des objets provenant de sources différentes mais équivalents d'un point de vue sémantique, ou encore complétées si des données supplémentaires (on parle de alors de métadonnées sémantiques) à celles déjà intégrées viennent décrire le contenu ou la sémantique des données déjà intégrées.

3.1.5 Le niveau de transparence

Le principal objectif des systèmes d'information est de rendre transparent l'utilisation des sources pour les utilisateurs. Un parfait système d'intégration donne l'illusion aux utilisateurs d'interagir avec un système unique et homogène. On distingue plusieurs niveaux de transparence :

Au niveau de la localisation Aucune information (nom, adresse) n'est nécessaire pour utiliser les sources de données à travers le système d'information.

Au niveau des schémas Les utilisateurs n'ont pas besoin de connaître les schémas définis sur les différentes sources de données et qui peuvent décrire une même entité biologique, et ses qualificatifs avec des termes différents.

Au niveau du langage L'interrogation du système ne nécessite pas d'expertise de la part des utilisateurs en ce qui concerne les langages de requêtes qui seront réellement utilisés sur les sources concernées.

Il y a clairement un lien entre le traitement de l'hétérogénéité et le niveau de transparence fourni par un système d'intégration de données. En effet, la transparence de schéma est fournie si le problème de l'hétérogénéité sémantique est résolu, alors que la transparence de langage et de localisation survient si on résout les problèmes liés à l'hétérogénéité syntaxique [Gué05].

3.1.6 Construction du schéma global d'intégration

On distingue deux manières de construire un système d'intégration : **top-down**, où l'on part de l'information souhaitée, pour ensuite chercher les sources pouvant répondre aux besoins, ou **bottom-up**, où l'on part de la volonté d'intégrer plusieurs sources de données.

Ainsi, dans les approches top-down, les schémas des sources importent peu pour la conception du schéma global. Ils seront seulement pris en compte dans un second temps quand les correspondances entre le schéma global et les schémas des sources seront établies pour permettre l'exécution de requêtes.

Dans l'approche bottom-up, il faut que le schéma global fournisse une vue conciliée des différentes sources, impliquant une bonne connaissance au préalable des schémas des sources de données.

3.1.7 Choix de la localisation des sources

Certains systèmes suivent une approche non matérialisée dans laquelle les données restent au niveau des sources et où les seules données matérialisées sont les résultats des requêtes au moment où elles sont exécutées. Ce type d'approche nécessite une transformation des requêtes posées au schéma global en une ou plusieurs requêtes qui seront distribuées dynamiquement aux sources concernées. Certains systèmes, au contraire, suivent une approche matérialisée, dans laquelle ils récupèrent les données partielles ou complètes des sources pour les stocker localement et les combiner dans un schéma global.

3.1.8 Langage de requêtes

Le langage de requête est le moyen avec lequel l'utilisateur accède aux données du système d'intégration. Il existe de nombreux langages de requêtes comme SQL, OQL, XQuery, etc., tous capturent l'expression du langage naturel avec plus ou moins de facilité et de détails. Les utilisateurs peuvent également accéder aux données par le biais de la navigation. De nombreux systèmes bioinformatiques basés sur le web, utilisent cette méthode.

3.2 L'approche matérialisée

3.2.1 Les entrepôts de données

L'entrepôt de données ou data warehouse, est un système informatique dont l'objectif est de centraliser un volume important de données consolidées[42] à partir des sources. En général, ils sont conçus pour que les utilisateurs accèdent rapidement aux informations stratégiques. Dans un entrepôt les données sont (i) nettoyées et homogénéisées, (ii) intégrées (provenance de plusieurs sources), (iii) datées (une indication sur leurs origines est conservée). Dans les entreprises, un data warehouse permet à ses utilisateurs de disposer d'informations pertinentes et d'outils d'analyse puissants pour faciliter la prise de décision. Le concept d'entrepôt se développe également le domaine de la biologie.

La différence entre une base de données et un entrepôt

Un entrepôt de données se distingue d'une base de données classique car ses fonctionnalités sont différentes [Fra97]. En effet, les bases de données offrent des opérations transactionnelles relatives au contenu des informations de la base alors que les entrepôts effectuent des processus analytiques. Les premiers systèmes sont d'ailleurs nommés OLTP (On Line Transactionnel Processing) et les seconds OLAP (On Line Analytical Processing). Les bases de données classiques sont caractérisées par : (i) un nombre important d'utilisateurs , (ii) des requêtes de sélection et d'insertion fréquentes, (iii) un traitement faible du volume de données au cours d'une transaction, (iv) le modèle de données est adapté pour minimiser les redondances tout en préservant la fiabilité et la cohérence des données.
Inversement, les entrepôts de données sont caractérisés par : (i) un nombre restreint d'utilisateurs, (ii) une grande variété de requêtes (simples et complexes) mettant en jeux (iii) un volume important de données, (iv) le modèle de données possède une structure complexe permettant des analyses dimensionnelle et des données historisées.

La définition d'entrepôt

Bill Inmon, un des précurseurs du concept de l'entrepôt de données, fournit la définition suivante [Inm96] : « Le data warehouse est une collection de données **orientées sujet**, **intégrées**, **non volatiles**, **historisées**, organisées pour le support d'un **processus d'aide à la décision**...Un data warehouse ne s'achète pas, il se construit. »

La figure 3.1 représente l'architecture d'un entrepôt de données. Avant d'être **intégrées** dans l'entrepôt, les données sont extraites des différentes sources par des extracteurs -adaptateurs (wrappers). Les wrappers ont plusieurs rôles. Spécifiques d'un type de source, ils en extraient les données, et effectuent des opérations de nettoyage et de transformation avant que les données ne soient chargées dans l'entrepôt. Toutes ces étapes se font dans une zone de préparation avant d'alimenter l'entrepôt. Les données intégrées sont archivées, datées, de cette manière elles sont **historisées**. Cette conservation de l'historique des données et de leur évolution permet d'effectuer des analyses comparatives et garantit des données **non volatiles**. La zone de présentation des données est chargée de répondre aux requêtes émises par les utilisateurs. Elle possède les

[42] La consolidation de données consiste au regroupement de différentes données statistiques (e.g. nombre d'occurrences trouvées), dans le but d'en accélérer l'analyse.

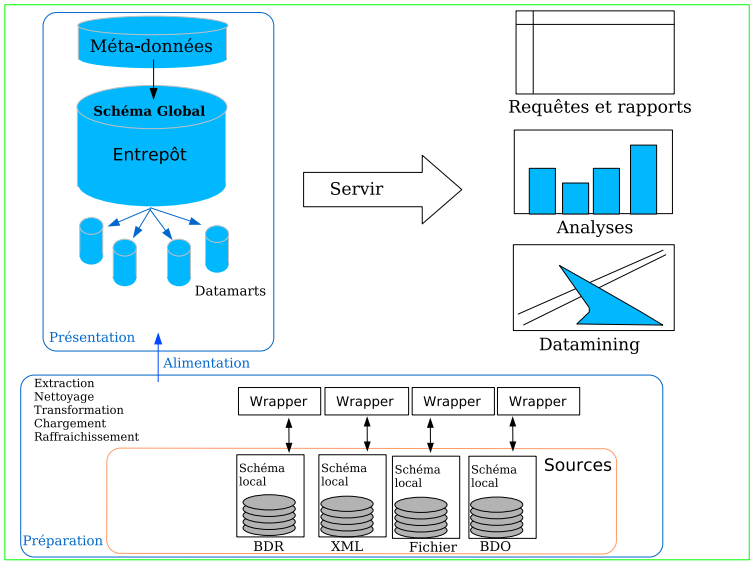


FIG. 3.1 – Architecture d'un entrepôt de données

outils et programmes nécessaires pour exécuter des requêtes complexes. Dans cette zone, les données sont regroupées dans l'entrepôt au sein de structures appelées les datamarts. Ces structures situées en aval de l'entrepôt représentent un point de vue sur un domaine ou un **sujet** donné. Les données stockées dans la zone de présentation servent à alimenter les outils de data mining et de visualisation.

Les spécificités propres aux entrepôts

Les entrepôts de données étant des systèmes décisionnels, le choix de leurs utilisations porte sur des besoins spécifiques. Nous allons décrire dans cette section leurs spécificités.

Spécificités de structure La principale caractéristique des entrepôts est d'organiser les données selon des axes d'analyses ou <dimensions>. Le modèle de données résultant est multidimensionnel. L'élément fondamental de ce modèle est le cube de données. Un cube organise les données en plusieurs dimensions. Une dimension correspond à un angle de vision porté sur les données. Un exemple de cube de données est représenté figure 3.2.

Le principe du cube de données est structuré selon différent modèles dans les entrepôts. On distingue les modèles en étoile, en flocon et en constellation. Tous ont en commun le même motif de base (e.g. l'étoile) dans lequel les dimensions encerclent l'élément de mesure (i.e. la variation du niveau d'expression dans notre exemple). Ces modèles sont implémentés dans des SGBD dont les types varient en fonction de leur capacité à gérer les données multidimentionnelles et leur rapidité (e.g. M-OLAP, R-OLAP, H-OLAP, O-OLAP).

Spécificité de fonction Avec leur modèle multidimensionnel, les entrepôts ont développé des opérations spécifiques pour manipuler les données. En plus des fonctions classiques que l'on retrouve dans les SGBRR (e.g. sélection, projection, produit cartésien), il y a des fonctions qui agissent sur la structure du cube ou qui agissent sur sa taille. Les fonctions les plus courantes (i) la rotation (slice) permet de changer l'orientation des axes de dimensions, (ii) l'extraction (dice)

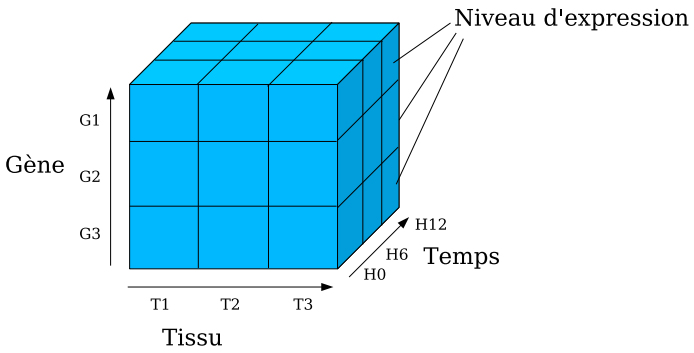


FIG. 3.2 – Exemple d'un cube de données. L'angle de vue dans cet exemple est la variation du niveau d'expression des gènes dans différents tissus au travers du temps. Le cube représente alors les trois axes : niveau d'expression génique, différents tissus et l'échelle du temps.

permet d'extraire une sous partie d'un cube, (iii) le forage vers le haut (drill-up) représente les données du cube à un niveau de granularité supérieur, (iv) le forage vers le bas (drill-down) représente les données du cube à un niveau de granularité inférieur.

Le data mining est un processus d'extraction de la connaissance à partir de grand volume de données. Il permet, sans intervention de l'utilisateur, de mettre en valeur des relations entre données, d'établir des statistiques, etc.

3.2.2 Les entrepôts de données en bioinformatique

Depuis les années 90, de nombreux avantages de l'approche entrepôt ont motivé son utilisation dans le secteur de la bioinformatique ([DCB+01, KKS+04, TRM+05, GMB+05, SHX+05]). En effet, certaines de leurs caractéristiques sont très bien adaptées aux problématiques de la bioinformatique. Par exemple, leur grande capacité de gestion et de stockage convient parfaitement à l'explosion de données que connaît le domaine. Les étapes nécessaires à l'intégration permettent de résoudre les problèmes d'hétérogénéité de plus la représentation multidimensionnelle des données est bien adapté à la complexité biologique. Enfin, au niveau de leur utilisation, ils permettent d'effectuer des requêtes complexes et performantes grâce à leur structure matérialisée et leurs applications spécifiques (e.g. data mining). La matérialisation des données permet aux utilisateurs de pouvoir faire des annotations.

Nous allons illustrer cette partie en présentant deux entrepôts décrits dans la littérature. Tous deux tentent de mettre à profit les caractéristiques de cette approche pour la génomique fonctionnelle. L'entrepôt de données GUS (Genomics Unified Schema) ([DCB+01]), à travers l'élaboration d'un schéma global générique et d'une intégration sémantique forte, fournit un environnement qui associe un gène à ses éventuels transcrits et protéines associés. GEDAW [GMB+05] est un entrepôt spécialisé dans les gènes responsables des maladies du foie chez l'homme. L'entrepôt développe sa spécificité sur l'intégration sémantique en utilisant les instances des sources pour effectuer les correspondances.

GUS

Genomics Unified Schema (GUS)[43] est un entrepôt de données conçu pour intégrer, analyser et représenter des données de génomiques fonctionnelles. GUS utilise un modèle relationnel afin de représenter des données génomiques, d'expression et de protéomique chez l'homme et la souris, mais le modèle est générique pour pouvoir être utilisé chez d'autres espèces [DCB+01]. L'important modèle (200 tables) possède une couche objet permettant une meilleure manipulation des données biologiques. Les données proviennent des principales banques de données internationales (e.g. Genbank, dbEST, Swiss-Prot, etc.). Les tables de GUS gèrent les annotations sur l'ensemble des séquences biologiques. GUS privilégie la traçabilité des données et la qualité et l'historique des annotations qui sont réalisées sur celles-ci. Le schéma de l'entrepôt représente ces notions et permet aux utilisateurs de filtrer les données avec ces critères (e.g. annotation manuelle, annotation automatique). Malgré son importante quantité d'information, GUS réalise ses mises à jour tous les deux ou trois mois. Chaque source est finement analysée pour ne modifier que les derniers changements. L'application composée du schéma, de l'interface web et des outils d'intégration est disponible librement ce qui en fait une application générique. De nombreux projets l'utilisent, comme par exemple, une base de données génomique, plasmoDB, ou une base de données d'expression, RAD.

GEDAW

GEDAW [GMB+05] est un entrepôt de données dédié à l'analyse des données d'expression des gènes du foie. Il intègre des données provenant de différentes expériences ou pathologies ainsi que des données issues des technologies à haut débit (Microarray, Macroarray, SAGE). L'objectif de GEDAW est de fournir des données intégrées de qualité permettant un support d'aide à la décision dans les études biomédicales. Les données intégrées proviennent principalement de sources structurées ou semi-structurées comme Genbank, GeneOntology ou UMLS. L'accent a été mis sur la qualité des données. Ainsi, lors de l'intégration de sources Genbank, les données sont nettoyées , annotées et complétées avec des informations provenant d'autres sources telles que LocusLink ou des liens avec L'UMLS ou Gene Ontology. Au niveau du nettoyage des données, GEDAW propose de regrouper les données similaires. Par exemple, deux séquences nucléiques qui sont étiquetées différemment mais qui ont des séquences identiques sont regroupées. Cette étape automatique est additionnée à l'expertise des chercheurs qui peuvent émettre des règles de nettoyage. L'intégration de données dans GEDAW s'effectue à deux niveaux : schémas et instances. Dans le premier cas, la source Genbank XML étant principalement utilisée, des programmes transforment les données vers un schéma global. Dans le second, les données sont regroupées afin d'éliminer des redondances ou clarifier des informations divergentes.

3.3 L'approche virtuelle

Dans cette approche, nous pouvons distinguer plusieurs degrés d'intégration (comme nous l'avions dit précédemment entre l'intégration à couplage lâche et celle à couplage serré).

En premier lieu, il faut considérer les portails, qui intègrent sur le même site web l'accès

[43] www.gusdb.org

à diverses sources. Ainsi, le portail Entrez[44] fournit l'accès à toutes les banques de données du NCBI (e.g. GenBank, UniGene, SNP etc.). De même, ExPASy[45] (Expert Protein Analysis System) [GGH+03], est spécialisé dans les sources protéomiques. Pour certains portails, il s'agit de proposer un point d'accès commun à des sources complémentaires. Mais dans ce type d'approche, l'interrogation des sources est limitée ([BBB+98, Kar95]). Même si les portails constituent des points d'entrée importants pour la communauté biologique, nous ne présenterons pas plus en détail leurs apports car pour la plupart d'entre eux l'intégration se borne à la présentation d'une collection de sources d'intérêt.

Dans un deuxième degré d'intégration, nous pouvons observer l'approche " navigationnelle " qui utilise les références entre données pour proposer aux utilisateurs une vision intégrée. Les architectures fédérées, quand à elles, proposent le regroupement de plusieurs bases de données hétérogènes et réparties. Dans cette approche, le degré d'intégration peut aller d'une simple correspondance entre les entités des sources jusqu'à la réalisation d'un schéma canonique. Enfin, nous plaçons en dernier la médiation de données, car les notions de schéma global, modèle de données et langage de requêtes commun sont implicites dans cette approche. Dans les sections suivantes, nous présentons deux types d'approches virtuelles fréquemment utilisées en bioinformatique, l'approche navigationnelle et la médiation.

3.3.1 L'approche navigationnelle

Principe de l'approche navigationnelle

Également appelée intégration à base de liens ou *link-based integration* (en anglais), elle est basée sur l'existence de relations entre les données présentes dans différentes sources du Web. Ces relations exploitent les caractéristiques du web et sont exprimées sous la forme de liens qui permettent la navigation internet à travers les sources (Figure 3.3).

L'idée de ce concept provient du besoin d'automatiser les recherches effectuées sur le Web par les biologistes [DOB95]. En effet, de nombreuses sources ne possèdent pas d'interface de requêtes évoluées et nécessitent une consultation manuelle des utilisateurs pour obtenir des informations. Le domaine de la bioinformatique possède une grande diversité de liens entre les sources de données. La prise en compte de ces derniers dans des applications permet également de découvrir de nouvelles relations entre les données ou des données exploitables par ce biais uniquement. D'après Hernandez et al. [HK04] l'approche navigationnelle n'implique pas une modélisation des données mais plutôt une modélisation de leurs relations, des points d'entrée, ainsi que des informations complémentaires telles que la spécification du contenu des sources, des éventuelles contraintes de chemins, et des paramètres facultatifs et obligatoires d'entrée. Les applications développant des approches navigationnelles doivent tenir compte de la diversité des liens existant entre les sources. En effet, outre les liens évidents que peuvent être une URL ou un lien hypertexte, il y a des liens implicites que l'humain peut détecter mais que les machines doivent reconnaître (e.g. AL082889 est un numéro d'accession GenBank correspondant à une séquence). On constate également que de multiples chemins (intermédiaires ou directs) peuvent lier deux sources. Nous allons le détailler dans le paragraphe suivant.

[44] http ://www.ncbi.nlm.nih.gov/Entrez
[45] http ://www.expasy.org/

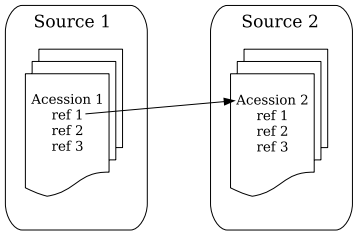


FIG. 3.3 – Liens entre deux accessions via une référence croisée. Chacune des sources contient des accessions comportant plusieurs liens de références croisées. Ces liens ne sont pas forcément bidirectionnels. Ainsi depuis l'accession 1 il est facile d'aller à l'accession 2. (Adapté de Guérin et al., [Gué05])

La diversité des chemins Lacroix et al. [LMNR04] montrent que pour une requête donnée, il existe un très grand nombre de chemins possibles au travers des sources. Pour l'illustrer, prenons l'exemple d'une requête « *lister toutes les citations de PubMed qui sont reliées à une entrée OMIM relevant de telle pathologie* ». Afin de trouver une réponse à cette requête, un biologiste (ou un moteur de requêtes) doit naviguer au sein de plusieurs sources. Le chemin le plus évident est de partir d'OMIM qui contient des informations sur les maladies génétiques humaines, puis d'utiliser la source PubMed du NCBI. La figure 3.4 illustre le graphe de liens existants entre les différentes sources du NCBI requises pour répondre à la requête.

Avec les sources disponibles sur ce graphe, on se rend compte que différents chemins sont possibles pour partir d'OMIM et accéder aux citations PubMed. Outre le chemin direct, il est également possible d'utiliser des sources intermédiaires, générant ainsi plusieurs chemins. Au total, cinq chemins existent entre OMIM et PubMed. Ils sont représentés sur la figure 3.5.

D'après Lacroix et al. [LMNR04], le choix des chemins a un impact sur le résultat, que ce soit sur le plan qualitatif ou quantitatif. Par exemple, utiliser un chemin passant par la source Protein (Figure 3.5, chemin C3) peut amener plus de citations qu'un autre chemin passant par la source GenBank (Figure 3.5, chemin C2). Le résultat va dépendre directement des sources intermédiaires du chemin et donc des entités biologiques correspondantes traversées et du contenu de chaque source.
On constate que la navigation à travers les sources apporte une grande richesse d'information. Cependant, cette approche nécessite une automatisation car la difficulté réside dans le temps de recherche effectué par les biologistes. Des systèmes d'intégration utilisant l'approche navigationnelle ont donc été développés.

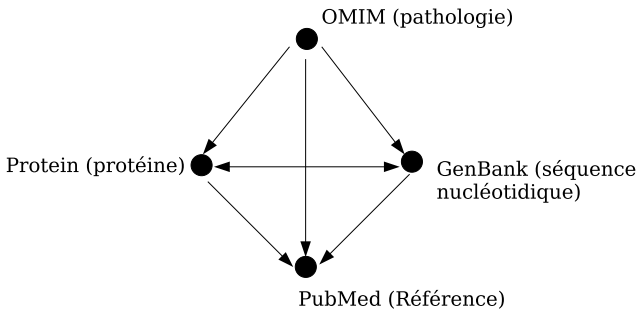


FIG. 3.4 – Graphe de liens entre les sources du NCBI (adapté de Lacroix et al. [LRV04]). Les points représentent les sources alors que les flèches représentent les liens entre les sources.

Systèmes bioinformatiques utilisant l'approche navigationnelle

De nombreux systèmes utilisant l'approche navigationnelle ont été développés en bioinformatique. Les différences varient selon le niveau de transparence des sources, l'évaluation de tous les chemins probables que peut générer une requête et la prise en compte des préférences utilisateurs pour l'exécution de la requête.

Le système SRS SRS (Sequence Retrieval System) est un système qui permet d'effectuer des recherches par mot clé à travers des banques de séquences. Initialement développé par l'EMBL puis par l'EBI [EA93, EUA96], SRS est maintenu depuis 1999, par LION Bioscience AG57. Il intègre aujourd'hui plus de 400 banques de données [ZLAE02].

L'approche d'intégration de SRS est basée sur un système d'indexation de sources de données structurées nommé ICARUS (Interpreter of Commands And Recursive Syntax). Ce langage permet de parcourir les fichiers structurés afin d'en indexer les données. Les index sont utilisés par la suite pour l'exécution des requêtes, les données n'étant plus nécessaires dans ce cas. ICARUS permet d'indexer les références croisées existantes dans les sources mais aussi d'en créer de nouvelles. Ainsi, l'indexation de plusieurs banques de données permet de créer un réseau de références croisées.

SRS possède une interface Web pour permettre aux utilisateurs de poser des requêtes. Cette dernière propose aux utilisateurs de choisir la source de données à interroger, ainsi que le mot clé ou la séquence à rechercher. Le résultat de la recherche correspond aux occurrences du mot clé ou de la séquence trouvés dans l'ensemble des banques sélectionnées. Un affichage plus détaillé de l'information permet de voir les données attachées ainsi que les références croisées.

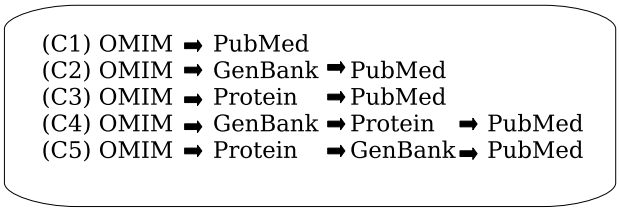
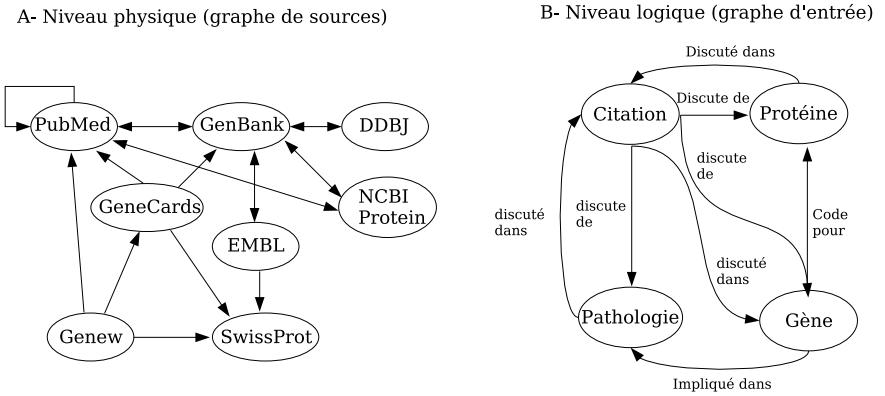


FIG. 3.5 – Illustration des chemins existants entre des sources. Les cinq chemins (C1 à C5) depuis OMIM jusqu'à PubMed en utilisant le graphe de la figure précédente (adapté de Lacroix et al. [LRV04])

SRS permet l'indexation d'une grande quantité de donnée et utilise les cross-références présentes dans les sources de données biologiques pour enrichir les requêtes. Toutefois le système n'exploite pas la diversité des chemins possibles pour l'exécution d'une requête. Ce sont Mork et al. qui ont proposé une approche transparente et qui tient compte des différents chemins générés pour répondre à une requête donnée [MHTH01]. Lacroix et al. ont ensuite introduit les défis d'estimation ([LMNR04] et d'optimisation des chemins [LRV04] en développant le système BioNavigation.

Le système BioNavigation Développé à l'université d'Arizona par Lacroix et al. [LMNR04, LRV04], BioNavigation est un système d'intégration basé sur l'approche navigationnelle. Toutefois, le système utilise des ontologies pour ajouter un niveau de transparence à l'utilisation des sources. D'après Lacroix et al., l'utilisation de telles ontologies permet d'avoir des requêtes plus ouvertes puisque les sources ne sont pas spécifiquement nommées.

La figure 3.6 est une représentation des ontologies de BioNavigation. On distingue tout d'abord le niveau physique (A) qui décrit les sources, leurs contenus et leurs liens entre elles, et le niveau logique (B) qui décrit les entités biologiques, les relations entre ces entités. BioNavigation fait également la correspondance entre le graphe des sources et le graphe des entités (C). L'ontologie est un support de requête pour l'utilisateur. Elle lui permet de naviguer au sein des différentes entités biologiques et de sélectionner graphiquement celles qui sont nécessaires à la construction d'une requête. Par exemple, un biologiste qui souhaite obtenir les références disponibles pour un gène donné, va sélectionner dans l'ordre, 'Gène', 'discuté dans' et 'Citation'. Le résultat se concrétise par une liste de chemins éventuels accompagnés de différents critères d'évaluation. Dans son principe, BioNavigation ne sélectionne pas le meilleur chemin , mais propose l'ensemble des chemins générés classés selon trois critères : la cardinalité du chemin (,nombre d'arcs séparant le point de départ et le point d'arrivée dans le graphe) la cardinalité de la cible (nombre de résultats disponibles dans la source finale) et le coût d'évaluation (coût total



C- Exemples de correspondances entre entités et sources

Entité	Sources de données
Citation	PubMed
Gène	GenBank; EMBL; DDBJ
Protéine	NCBI Protein; SwissProt

FIG. 3.6 – Niveaux de représentation dans BioNavigation et correspondances entre entités biologiques et sources de données (Adapté de [LMNR04]). Le niveau physique (A) représente les liens qui existent entre les différentes sources. Le niveau logique représente les liens qui existent entre les différentes entités biologiques. Le niveau C représente les correspondances entre entités et sources des deux niveaux précédents.

de l'exécution). Grâce à ce classement le biologiste peut sélectionner le chemin qu'il estime le meilleur.

Le système BioGuide Le système BioGuide utilise une approche navigationnelle qui prend en compte les préférences utilisateurs dans ses stratégies de recherche [CBDF+06, CBFP06]. En effet, sur la base d'enquêtes, les auteurs, ont démontré que les scientifiques expriment des préférences concernant le choix des sources à interroger et des outils à utiliser. De plus, ils utilisent des stratégies différentes selon leurs types de recherche. Grâce aux enquêtes, 30 critères identifiés servent à définir la requête utilisateur. Un des critères définis est la fiabilité des sources qui peut être ajustée selon ses propres préférences (e.g. mettre un seuil de confiance de 9 sur 10 à la source SWISS-PROT). Les stratégies de recherches peuvent également être définies par l'utilisateur. Elles concernent par exemple l'ordre dans lequel il désire parcourir les sources.

Le système BioGuide possède une interface qui permet aux utilisateurs de définir leurs propres requêtes, de régler divers paramètres liés à leurs préférences mais également leurs stratégies d'exécution (dans les deux cas des valeurs par défaut sont proposées). BioGuide peut être utilisé de manière générique ou bien à l'interface du système SRS. Dans ce cas, il permet aux utilisateurs de construire de puissantes requêtes sur la quantité importante de sources que contient SRS.

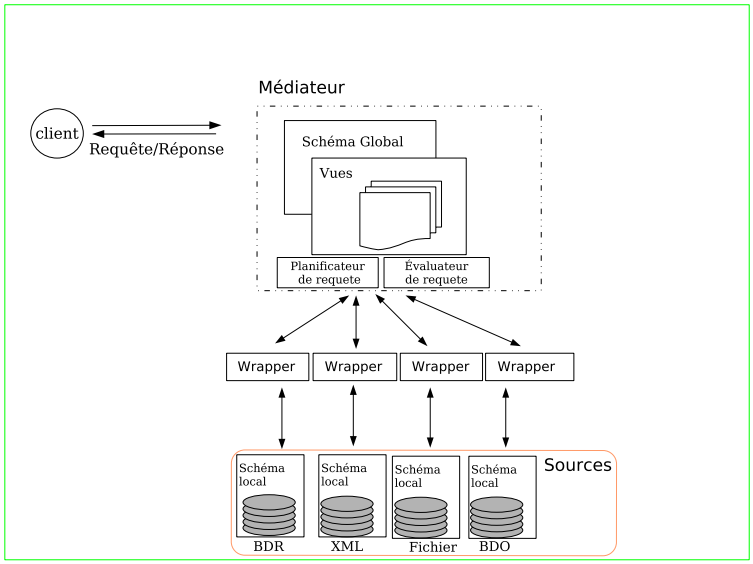


FIG. 3.7 – Architecture d'un médiateur

3.3.2 La médiation

Principe de l'approche

Wiederhold introduit la notion de médiation comme une interface virtuelle réconciliant des sources de données distribuées, autonomes et hétérogènes [Wie92]. Les utilisateurs qui interagissent avec cette dernière posent des requêtes de manière transparente sur les sources distribuées en ayant l'impression d'interroger un système centralisé et homogène.

Un système de médiation est composé d'un médiateur et de plusieurs adaptateurs (ou *wrappers*) qui sont spécifiques d'une source donnée. La figure 3.7 représente une architecture médiateur standard. Nous pouvons distinguer trois niveaux dans cette architecture : le wrapper, le médiateur et le client.

- L'adaptateur ou *wrapper* est un programme qui se place à l'interface entre le médiateur et la source. Le *wrapper* propose une vue de la source dans le modèle de représentation du médiateur. Il sert également de traducteur entre le langage de requête de la source et celui du médiateur. Enfin il transforme les résultats exprimés dans le modèle de données de la source vers le modèle de données du médiateur. Certains *wrappers* peuvent envoyer des messages au médiateur tels que ceux relatifs à la disponibilité de la source, au temps estimé d'exécution, etc.
- Le médiateur centralise et intègre les vues des sources de données disponibles. L'intégration s'effectue à travers un schéma global. Les utilisateurs interrogent le schéma global. Le rôle du médiateur est de réécrire cette requête, l'exécuter et retourner les résultats. Pour cela, il identifie tout d'abord, quelles sont les sources pertinentes pour y répondre. Dans une deuxième phase, il décompose la requête de manière à ce que chaque sous-requête puisse être envoyée à une des sources pertinentes détectées et établit un plan d'exécution. Ces tâches sont gérées par l'évaluateur et le planificateur de requêtes. Enfin, il reconstruit le résultat de la requête globale à partir des résultats des sous-requêtes.
- Le client est une application susceptible d'interroger le schéma global du médiateur. Cela peut être un navigateur web, une application, une interface graphique, etc.

Approches de conception du schéma global

Un moyen de comparer les architectures de médiation est la manière dont est conçu le schéma global [Lev99]. Nous présentons les deux principales approches.

L'approche **GAV** (**Global As View**) [Hal01] définit le schéma global comme une vue sur les schémas des sources. Dans ce cas, le traitement des requêtes est facilité par la correspondance claire des parties du schéma global avec les schémas des sources. Ce type d'approche est bien adapté aux sources complémentaires. Par contre, il est plus complexe d'ajouter de nouvelles sources " chevauchantes " au système car cela pose des problèmes d'hétérogénéité sémantique.

Dans l'approche **LAV** (**Local As View**) le schéma global est défini indépendamment des sources. En quelque sorte, il modélise le domaine concerné. Pour réaliser la correspondance avec les sources, des vues sont réalisées à partir de ces dernières. Avec cette approche, l'étape de correspondance (ou *mapping*) entre le schéma et les sources est plus complexe. En revanche, l'ajout d'une nouvelle source est facilité car le schéma global reste inchangé.

La médiation est largement utilisée comme approche virtuelle en bioinformatique. Dans la section suivante, nous en donnons un aperçu en présentant quatre architectures : K2, TAMBIS, BACIIS et DiscoveryLink.

3.3.3 Systèmes bioinformatiques utilisant l'approche de médiation

K2/Kleisli

A l'origine, BioKleisli [DOTW97] est un système de médiation de sources bioinformatique qui utilisait un langage de requête dédié : le CPL (Collection Programming Language) [HSO94]. Utilisé pour son expressivité, ce langage permet de réécrire les requêtes utilisateurs en plusieurs sous requêtes. La nouvelle version de BioKleisli, K2, [DCB+01], utilise le langage de requête OQL (Object Query Language) et un modèle de représentation objet. Le système intègre de nombreuses sources. Parmi elles, les données sur les voies métaboliques de KEGG[46] (Kyoto Encyclopedia of Genes and Genomes) [KG00] et EcoCyc[47](Encyclopedia of Escherichia coli) [KCVGC+05], les banques de séquences nucléiques de GenBank et dbEST[48] (Expressed Sequence Tags database) [BLT93], des données spécifiques d'organismes de MGD et de GDB[49] (human Genome DataBase) [PMFR92], ainsi que des données issues de programmes tels que BLAST [AGM+90].

De récentes améliorations de K2 permettent aux utilisateurs de définir des vues sur les données. En plus des caractéristiques d'OQL dans ce domaine, les auteurs ont développé le langage K2MDL, une combinaison du langage ODL (Object Definition Language) et de la syntaxe de OQL, qui permet aux utilisateurs , de créer de nouvelles classes en effectuant un *mapping* des attributs avec les sources. K2 introduit une manière différente d'exécuter des requêtes, puisque ces objets créés peuvent être interrogés par OQL. Cette notion se retrouve également dans TAMBIS.

[46]http ://www.genome.ad.jp/kegg/
[47]http ://ecocyc.org/
[48]http ://www.ncbi.nlm.nih.gov/dbEST/
[49]http ://gdbwww.gdb.org/

TAMBIS

Le système TAMBIS (Transparent Access to Multiple Biological Information Sources) est un projet de recherche, développé à l'université de Manchester, dont l'objectif est de fournir un système transparent de requêtes sur des sources distribuées et hétérogènes (Baker et al., 1998 [BBB+98]). L'originalité de TAMBIS est qu'il utilise une ontologie comme schéma de médiation, Tambis Ontology [BGB+99] ou TaO. L'ontologie joue deux rôles. Elle sert de support pour la formulation des requêtes utilisateurs en effet elles sont exprimées à partir des termes plutôt que directement sur les sources. Elle joue également un rôle de médiateur entre les sources, en réglant les conflits sémantiques au niveau de l'abstraction. Pour décrire les concepts biologiques, TaO reprend les travaux du projet GALEN [RBG+97]. Ce projet propose une logique de description GRAIL ainsi qu'une ontologie [RN94]. TaO a étendu cette ontologie pour qu'elle intègre les sources bioinformatiques.

L'ontologie peut être divisée en deux parties. Une hiérarchie de concept de haut niveau d'abstraction utilisé dans le projet GALEN. Ce premier niveau fait la distinction entre le domaine des concepts et le domaine des rôles. Dans ce premier domaine sont décris des concepts généraux traitant des structures, substances, process et fonction. Dans le domaine des rôles sont définis des relations génériques telles que la collection et la localisation. Cette hiérarchie de haut niveau a été étendue dans TAMBIS avec une hiérarchie de bas niveau représentant les connaissances utilisateurs dans le domaine biologique. Ce modèle est centré sur la description des concepts Protéines, Acides nucléiques et les concepts qui en découlent comme enzyme, DNA et RNA . Les fonctions et process biologiques sont aussi décrits. Tous ces concepts sont présents dans une hiérarchie de type " is a kind of " mais d'autres relations comme "is component of " enrichissent le modèle grâce aux rôles. 1800 concepts sont décrits traitant des séquences protéiques de SwissProt, des motifs et structures de Prosite, des enzymes et voie métabolique de Enzyme DB, des EST de DB EST. Des concepts sont aussi construits sur les homologies de séquence et sur la taxonomie définie selon le NCBI.

Lorsqu'un utilisateur pose une requête, il navigue dans l'ontologie et sélectionne les concepts et rôles nécessaires à la formulation de sa requête. Par exemple, pour sélectionner un type particulier de motif dans une protéine, les concepts "Motif" et "Protein" ainsi que le rôle "IsComponentOf" qui les associe sont sélectionnés ; un nouveau concept est construit automatiquement dans le langage GRAIL. Ce nouveau terme est automatiquement classé dans la hiérarchie de concepts de TaO. Par la suite, le processeur de requête établit un plan d'exécution et transforme la requête GRAIL en sous-requêtes à soumettre à des sources différentes. Ces requêtes sont converties dans le langage CPL [HSO94], correspondant aux différentes sources de données. Le résultat du plan de requêtes est ensuite délivré à l'utilisateur au format HTML.

Récemment, TaO a été traduite dans le langage DAML+OIL [SGHB02], puis OWL qui sont des langages plus expressifs. De cette manière, TAMBIS fournit un accès transparent aux sources de données. De plus, les utilisateurs peuvent accéder aux sources sans maîtriser un langage de requête particulier.

BACIIS

Le système BACIIS (Biological and Chemical Information Integration System) est un médiateur qui utilise une ontologie (BAO) comme schéma de médiation [MKL+05]. Comme dans le cas de TAMBIS, elle permet de résoudre les problèmes d'intégration sémantique et sert de support à la définition de requêtes utilisateurs. BACIIS se distingue par un grand nombre de

sources intégrées, particulièrement des sources chevauchantes. Les concepteurs du système considèrent que ce type d'intégration permet d'augmenter la pertinence des résultats. En effet, le rôle de BACIIS est de fournir des solutions à l'incomplétude des sources ou à d'éventuels conflits entre données. Dans ce domaine, le système évalue la correspondance sémantique entre deux objets de sources différentes au moment de l'intégration afin d'éliminer les données sémantiquement distantes.

DiscoveryLink

DiscoveryLink [HSK+01] développé par IBM est un système d'intégration bioinformatique basé sur des wrappers. L'objectif de DiscoveryLink est de fournir une couche d'échange et de communication entre des applications et des sources distribuées. De cette manière les applications se connectent au middleware et soumettent une requête SQL au schéma global sans nécessairement avoir connaissance des sources sous-jacentes.

DiscoveryLink est essentiellement une couche d'intégration construite sur la technologie du projet Garlic [RAH+96, CHN+95]. La technologie de Garlic est un moteur de requête de bases de données fédérées capable de communiquer avec des wrappers spécifiques pour une source pour définir un plan optimal de requête et exécuter cette dernière. Le rôle des wrappers est très important puisque en plus de jouer un rôle de traducteur entre le middleware et la source, ils vont communiquer des méta-informations en termes de coûts (temps de calcul, temps d'accès , temps d'exécution, etc).

DiscoveryLink utilise un modèle objet-relationnel comme modèle pivot. A l'inverse de TAMBIS, DiscoveryLink n'est pas un développé avec une interface de consultation. Une interface est donc nécessaire pour opérer au dessus du middleware, construire les requêtes et les exécuter. Ils diffèrent également dans l'optimisation du plan de requête et le choix des wrappers. En effet, alors que TAMBIS préférera une optimisation basée sur la sémantique de la requête, DiscoveryLink sera centré sur les performances par l'utilisation de ses wrappers.

3.4 Discussion

Nous avons décrit dans la section précédente, les différentes approches d'intégration utilisées en bioinformatique à travers les approches matérialisées et virtuelles. Pour chaque type d'approche d'intégration, différents systèmes ont été présentés reflétant ainsi l'état actuel des développements dans le domaine de l'intégration de données en bioinformatique. Pour chaque système, nous avons tenté de mettre en évidence leurs caractéristiques par rapport aux critères d'évaluation des systèmes d'intégration, tels que décrits dans la section 3.1. Le tableau 3.8, résume et illustre ces caractéristiques.

Les principaux avantages de l'approche matérialisée proviennent du traitement des requêtes et de l'accès en écriture sur les données intégrées. La facilité d'optimisation des requêtes s'explique par le fait que les données sont centralisées localement et unifiées dans un schéma global. L'accès en écriture permet aux utilisateurs d'annoter les données. Ce point qui est illustré dans l'entrepôt GUS (section 3.2.2), même s'il est très avantageux, présente des inconvénients dans la mesure où il est très coûteux en temps de maintenance et complique la tâche de mise à jour du système. Toutefois, ce point n'entache pas les qualités des entrepôts qui sont capables de

prendre en compte efficacement l'inconsistance des données provenant de différentes sources, et qui fournissent des moyens d'analyses avancés sur de grands volumes de données.

Les approches virtuelles sont mieux adaptées aux analyses ponctuelles, sur de faibles volumes de données. Elles ont l'avantage de ne pas stocker les données localement et donc de disposer de données à jour. Leur faiblesse, se situe au niveau du temps d'exécution des requêtes car les systèmes sont très dépendants de la disponibilité et de l'accessibilité de ces sources externes. Parmi les approches virtuelles, on distingue celles qui utilisent des ontologies, telles que TAMBIS, pour concevoir leur schéma global et y effectuer des requêtes en mettant en œuvre des stratégies d'interrogation.

Dans ce domaine, BioGuide propose à l'utilisateur un paramétrage de ses préférences tandis que BioNavigation propose une sélection de critères pour évaluer la satisfaction d'une requête. Une des caractéristiques propres à la bioinformatique est la prise en compte de l'intégration navigationnelle à travers l'optimisation des chemins (cf. BioNavigation).

On constate que la plupart des approches virtuelles n'effectuent qu'une intégration horizontale des données en intégrant uniquement des sources de données complémentaires. Dans ce domaine, l'approche matérialisée propose des solutions afin de résoudre les problèmes liés aux données absentes ou contradictoires, et identifier les données de mauvaise qualité.

Application	Données Intégrées	Type d'intégration	Modèle de données	Intégration sémantique	Niveau de transparence	Shéma global	Type d'approche	Langage pivot
Gus	Tout Formats	Tight	Structurées, relationnel	Données complémentaires	Totale	Bottom-up	Matérialisée	SQL+Web
GEDAW	Tout Formats	Tight	Structurées, objet	Données chevauchantes	Totale ou choix des sources	Bottom-up	Matérialisée	OQL
SRS	Tout Formats	Loose	Fichiers plats	Données complémentaires	Schéma	Top-down	Virtuelle	Navigation Web
BioNavigation	Tout Formats	Tight	Structurées, relationnel-objet	Données complémentaires + chevauchantes	Totale ou choix des sources	Top-down	Virtuelle	Requête via un graphe
BioGuide	Tout Formats	Tight	Structurées, relationnel-objet	Données complémentaires + chevauchantes	Totale ou choix des sources	Top-down	Virtuelle	Requête via un graphe
K2	Tout Formats	Loose	Structurées, objet	Données complémentaires	Pas de sélection des sources	Top-down	Virtuelle	OQL
TAMBIS	Tout Formats	Tight	Structurées, relationnel-objet	Données complémentaires	Totale	Top-down	Virtuelle	CPL
BACIIS	Tout Formats	Tight	Structurées, relationnel-objet	Données complémentaires	Totale	Top-down	Virtuelle	Web
DiscoveryLink	Tout Formats	Tight	Structurées, relationnel	Données complémentaires	Totale	Top-down	Virtuelle	SQL

FIG. 3.8 – Table récapitulative des systèmes d'intégration mettant en valeur les critères décrits en section 3.1

Deuxième partie

Propositions : intégration de ressources végétales

Chapitre 4

Premier pas vers l'intégration

Sommaire

4.1 Introduction

AFIN de contribuer à l'effort international en matière d'analyse fonctionnelle du génome du riz (*Oryza sativa*), notre laboratoire (UMR PIA 1096) a produit une collection de 30 000 lignées d'insertion T-DNA dans le cultivar séquencé Nipponbare [SGL$^+$04]. Chaque lignée intègre en moyenne 2,2 copies du T-DNA. En plus de l'altération de la fonction génique, le T-DNA est équipé d'un gène rapporteur *GUSA* ou GAL4 :GFP [JHG$^+$05] ainsi que d'un élément activateur d'expression qui permet l'observation d'une activité GUS ou d'une fluorescence GFP dans le tissu ou l'organe muté. Par ailleurs, la transformation des plantes par culture *in vitro* peut provoquer d'autres événements de mutation. En effet, la transformation peut induire la transposition du rétro-élément *Tos*17, avec à l'arrivée 3,4 copies en moyenne par lignées T-DNA [SMvB$^+$03].

Depuis 2002, un effort important de caractérisation moléculaire et morpho-physiologique a été effectué sur cette collection. Cet effort inclut la détection des sites d'insertion (T-DNA et *Tos*17), dans chaque lignée, l'évaluation au champ de 13,928 lignées et la description pour les caractères morpho-physiologiques de leurs descendances. Une partie de la collection a été spécialement étudiée pour les caractères spécifiques du grain ainsi que pour la réponse à l'infection par le champignon *Magnaporthe grisea*. Les tissus ou les organes spécifiques dans lesquels l'expression des gènes rapporteurs GUS et GFP est détectée ont été déterminés.

Les objectifs poursuivis étant de nous consacrer à l'analyse fonctionnelle du génome du riz, plante modèle pour les céréales, il s'agit donc de proposer une organisation de cette information et de la rendre accessible par Internet à la communauté internationale.

Des premiers pas ont consistés à développer deux systèmes d'information : Oryza Tag Line et OryGenesDB. Le premier contient des données morpho-physiologiques, des informations générales sur la collection ainsi que des données d'expression. Le deuxième est dédié à la génétique inverse, c'est-à-dire qu'à partir d'un gène, il permet de remonter jusqu'à la plante mutée. Le système contient des données génomiques et l'information sur les sites d'insertion des T-DNA : les FST (Flanking Sequence Tags). Il intègre toutes les informations de séquences et d'annotation susceptibles d'enrichir l'information sur la fonction des gènes. Dans la suite de ce chapitre, nous allons détailler comment ces deux systèmes ont été conçus en suivant une approche intégrative explicite, puis comment est pensé leur fonctionnement en montrant, au-delà du premier effort d'intégration, qu'ils constituent des candidats à une intégration ultérieure que nous serons amenés à proposer et à détailler dans les chapitres suivants.

4.2 Oryza Tag Line

4.2.1 Matériels et méthodes

4.2.1.1 Conception et mise en oeuvre

Oryza Tag Line est un système d'information regroupant des informations issues de plusieurs domaines d'étude des plantes, de la collection d'insertion T-DNA. Son modèle réconcilie différents points de vue dans le but de créer une ontologie du domaine. D'ailleurs le modèle aborde et s'appuie sur la notion de vocabulaire contrôlé (et par extension les ontologies biologiques du même format que Gene Ontology) afin de créer des correspondances entre ces différents points de vues.

Le modèle conceptuel de la base de données a été réalisé dans un diagramme de classes sous le formalisme UML[50]. Il contient 9 packages : *contact, reference, phenotype, insert, stockage, expression, line, ontology et manage. Contact* permet de gérer les personnes interagissant avec le projet (par exemple les fournisseurs de données, les demandeurs de lignées, etc.). *Reference* permet de gérer les informations de bibliographie liées aux données (par exemple, la référence bibliographique d'un gène muté). *Phenotype* modélise la gestion des observations phénotypiques. *Insert* gère les informations liées à l'identification des insertions (e.g. T-DNA, *Tos*17). *Stockage* permet de gérer les stocks de graines produites par les plantes. *Expression* modélise les données d'expression. Line représente les relations des lignées avec leur environnement (par exemple, des conditions de culture, leur localisation, etc.). *Ontology* modélise la relation des termes issus des vocabulaires contrôlés avec les données de la base. *Manage* modélise la production des lignées d'insertion. La figure 4.1 représente le package Line qui est implémenté dans la base de données.

Compte-tenu d'une part de la confidentialité des données produites et d'autre part du manque de connaissance de la communauté en termes de langages d'interrogation, l'interface d'accès proposée est conviviale, propose trois niveaux de connexion (public, privé, expert) et une interrogation aisée via des formulaires appropriés.

Le modèle a été implémenté sur un système de gestion de bases de données relationnelles Oracle v8i (tables relationnelles et vues ont été déclinées). Pour la consultation des données de la base, une interface de consultation a été programmée en HTML et Perl CGI. Pour le chargement des données dans la base, des programmes spécifiques ont été conçus. Les données ne sont chargées que par l'administrateur, ce qui évite des conflits lors de l'insertion ainsi que les problèmes de transaction. Des API perl spécifiques ont été développées pour extraire les données stockées dans des formats hétérogènes. Par exemple, les observations phénotypiques sont stockées dans des fichiers Excel alors que les données d'expression sont accessibles à partir d'une application FileMaker. L'API est développée avec une structure modulaire. Les fonctions développées permettent (i) d'extraire des données en fonction du type de source et du schéma de la source, (ii) d'uniformiser les syntaxes, (iii) de vérifier la cohérence des données, (iv) de créer des liens avec les images, (v) enfin de créer les index.

4.2.1.2 Contenu du système

La base de données contient actuellement 30 000 lignées dont 13 928 lignées ayant un stock de graines suffisant pour être distribué. Les caractérisations sont en cours pour atteindre pro-

[50] http ://www.uml.org/

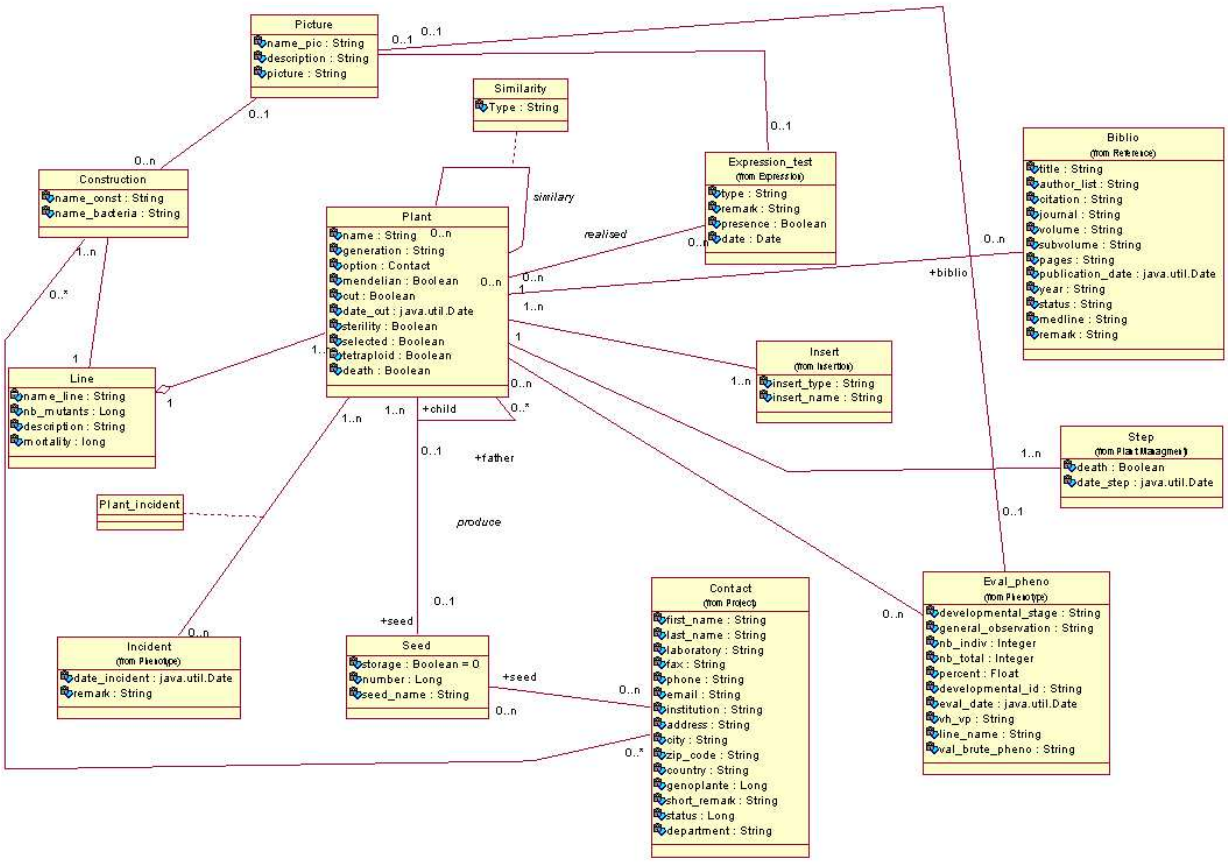


FIG. 4.1 – Représentation du package Line dans un diagramme de classes sous le formalisme UML

chainement le nombre des 30 000. En plus des données phénotypiques et d'expression, décrites ci-dessous, des informations utiles sont liées à chaque lignée. Par exemple, il est possible de visualiser un graphique de la construction génétique du T-DNA inséré par transformation dans la plante. Pour les lignées décrites dans la base, une nomenclature a été mise en place qui permet de savoir de quelle génération il s'agit. Les plantes ayant été directement transformées par le T-DNA sont nommées T0 alors que leurs descendances sont nommées T1.

Données phénotypiques (i) L'observation de panicules portant les grains T1 (issus de la première génération) sur 7187 lignées T0 (transformants primaires) permet d'évaluer 251 phénotypes mutants soit 3,5%. Les altérations observées portaient sur des grains avortés, ridés, réduits ou déformés. (ii) La réponse à l'infection par *Magnaporthe Grisea* a été effectuée à partir de plantules issues de 4462 transformants primaires. L'infection a été réalisée au stade 4-5 feuilles avec des spores du champignon. Les analyses ont été effectuées 5 jours après l'infection pour évaluer la résistance ou la sensibilité au champignon. Les résultats montrent que 44 lignées (1%) affichent une augmentation ou une diminution de la sensibilité et que 69 (1,5%) sont résistantes. (iii) Pour l'évaluation au champ des lignées, les caractères morpho-physiologiques ont été relevés pour 25 descendants (T1) des transformant primaires (25 plantes T1 par T0). Les évaluations phénotypiques ont été effectuées toujours aux mêmes stades de développement. Une première évaluation est effectuée 45 jours après germination, puis une autre au stade de la floraison et, éventuellement, une dernière à maturité. En tout, 258 descripteurs phénotypiques semi-quantitatifs ont été observés et classés en 6 classes de caractères incluant la morphologie, la phyllotaxie, la physiologie, la pigmentation et les caractères des panicules. Comme pour les études faites précédemment chez Arabidopsis, seule une faible fréquence (5-10%) des phénotypes sont corrélés avec la présence de l'agent mutagène (T-DNA ou *Tos*17). Sur l'ensemble de la collection, le travail sur la résistance à *M. grisea* montre un taux d'étiquetage de 10%.

Données d'expression GUS et GFP GFP Les essais d'expression des gènes rapporteurs GUS et GFP ont été réalisés dans [SMvB$^+$03] et [JHG$^+$05] respectivement. Pour les essais d'activité GUS, l'équipe a systématiquement testé les tissus des feuilles et fleurs des T0 ainsi que la moitié d'un grain mature T1. Pour les essais GFP, en plus des mêmes essais que pour GUS ont été conduits, mais les tissus des racines et tiges des plantules T1 3 et 5 jours après germination ont également été testés.

Information FST (Flanking Sequence Tag) Les régions flanquantes des insertions (FST) sont identifiées par séquençage pour les deux coté du T-DNA et pour le 3'LTR du Tos17. Cette information est stockée dans la base d'OryGenesDB mais un lien permet d'établir une référence croisée avec OTL (Partie B et E de la figure 4.2). Au total, 8 004 et 6 101 des 13 928 lignées sont caractérisées par au moins un insert T-DNA ou *Tos*17.

4.2.2 Résultats

4.2.2.1 Analyses des données

La vision intégrée du modèle conceptuel a permis de rapprocher des informations provenant de sources différentes, dégageant ainsi de nouvelles relations entre les données et de nouvelles connaissances.

Actuellement, OTL intègre des données provenant de 13 928 lignées sur un total de 30 000 évaluées au champ. OTL stocke 9721 enregistrements de mutants évalués pour 6 grandes classes de caractères. 2 636 lignées soit 19% ont été observées pour une mutation ce qui est le taux attendus pour une ségrégation Mendélienne sur un simple locus. Parmi ces dernières 30,1, 18,2 et 25,6% sont caractérisées par au moins un T-DNA, au moins un Tos17 ou les deux. Les altérations les plus fréquentes dans la collection portent sur la hauteur des plantes et la stérilité male. Le caractère albinos représente 7,6% des phénotypes observés. Les caractères ont été annotés avec les termes de la Plant Ontology lorsqu'ils correspondaient parfaitement [JAI+05, WJN+02].
Un total de 27 et 29% des lignées testées pour les tissus végétatifs et floraux affichent une activité GUS ou GFP respectivement dans au moins un des tissus ou organes.

4.2.2.2 L'interface du système

Oryza Tag Line est accessible à l'adresse http ://urgi.versailles.inra.fr/OryzaTagLine/. Les utilisateurs peuvent effectuer des recherches de plusieurs manières.

Recherche par phénotype Cette interface permet d'extraire une liste de lignées ayant des caractéristiques spécifiques. Par exemple elle peut être effectuée en recherchant les altérations observées à un stade de développement précis (e.g. tillering), ou dans un organe spécifique (e.g. leaf) ou encore pour un caractère particulier (e.g. morphology). Des critères restrictifs permettent de lancer une recherche pour les seules plantes ayant des graines disponibles ou des FST identifiées.

Recherche par mot clé Elle permet d'effectuer une recherche libre sur tous les champs qui ont trait aux observation phénotypiques (par exemple, nom de mutant, caractère, description phénotypique, synonymes, abréviations). Le résultat de la recherche génère une liste de lignées correspondant aux critères de départ.

Recherche par expression La recherche par expression permet d'extraire une liste de lignées ayant des observations pour les expressions recherchées. Comme pour les recherches précédentes des critères fins peuvent être appliqués (par exemple, organe, tissus, type d'expression , etc). Des contraintes fortes peuvent être appliquées, par exemple une recherche d'expression dans un organe spécifique.

Recherche avancée Ce type d'interface permet de mélanger deux types de recherches distinctes : phénotype et expression. Cette interface est conçue pour extraire les lignées ayant plusieurs observations dans des domaines multiples (par exemple, phénotypique, expression, FST, etc.)

Recherche par vocabulaire contrôlé Un grande partie des données stockées dans la base de données ont été annotées selon plusieurs vocabulaires contrôlés [HCI+04, IKJ+07, JAI+05, PJK+06, WJN+02, YJ05] (e.g. trait, plant structure, cereal growth stage et gene ontology). Concrètement, les données ont été indexées avec les ID les identificateurs des différentes ontologies (e.g. le terme "ligule length" correspond au TO :0000024 de la Trait Ontology). Cependant, il y a encore de nombreuses données qui ne correspondent pas aux termes des vocabulaires contrôlés. L'interface correspond à une recherche par identifiants ou termes, ou alors une navigation dans les taxonomies de vocabulaires contrôlés. Le résultat se présente sous la forme d'une liste de lignées possédant le critère annoté.

La figure 4.2 illustre une recherche de mutant dans Oryza Tag Line. Toutes les interfaces de recherche décrites ci-dessus permettent d'obtenir une liste de lignées similaire à celle qui est

représentée dans la partie A. Dans ce cas, cette liste est le résultat d'une recherche d'expression GFP dans les fleurs. Les résultats indiquent également la présence de FST afin de pouvoir rechercher, dans un second temps, des informations sur les gènes responsables des mutations. La sélection d'un lignée particulière permet d'afficher des informations générales (partie B). De nombreuses informations sont disponibles entre autre la disponibilité de graines, la présence de FST ainsi que toutes les observations réalisées sur la lignée. A partir d'un lien de référence croisée, il est possible de voir la position du site d'insertion de la FST dans le génome du riz (via OryGenesDB) et donc de connaître les gènes proches de l'insertion (partie E). Par ailleurs, il est possible de visualiser les observations détaillées (partie C et D).

4.2.3 Discussion

Avec les 13 928 lignées caractérisées pour 266 caractères phénotypiques d'intérêt, Oryza Tag Line représente, avec les bases de données Tos17 du NIAS [MIK+07] et de T-DNA RMD [ZLW+06], une ressource très utile pour la recherche en génomique fonctionnelle. De nouvelles observations sont continuellement intégrées dans la base, l'objectif étant d'atteindre les 30 000 lignées annotées.

Sur un plan technique, le système doit évoluer sur deux points importants : la généricité de son modèle afin qu'il prenne en compte les besoins de la communauté graminées voir plantes et l'évolution de son interface de consultation pour qu'elle s'adapte aux préférences des utilisateurs.

Le modèle conceptuel évolue constamment afin de prendre en compte les nouveaux types de données à intégrer. Développé dans un premier temps, spécifiquement pour des projets d'analyse chez le riz, le modèle doit être éprouvé de manière générique. Dans ce domaine, nous pouvons nous inspirer des deux projets Chado [MEC07] et CGP [BDH+06] qui proposent des modèles et des outils intéressants et "open source". De plus, l'utilisation de tels systèmes peut faciliter le partage d'information entre applications du même type.

Au niveau de l'interface Web, il est possible de sauvegarder les résultats dans un fichier Excel mais les utilisateurs ont besoin d'avoir un espace de travail virtuel dans lequel ils peuvent stocker leurs différentes recherches et les manipuler (par exemple, fusionner, effacer, etc.). Afin de partager et mutualiser le travail, cet espace de travail doit prendre en compte la notion de groupes dans lesquels sont partagées les données.

4.3 OryGenesDB

4.3.1 Matériels et méthodes

4.3.1.1 Conception et développement

OryGenesDB est né de la volonté de plusieurs équipes du CIRAD, de manière à proposer des interfaces graphiques et des outils d'analyse adaptés à la recherche de mutants par génétique inverse.

Les données concernées ont leur origine détaillée dans la figure 4.4. Nous retrouvons des données issues des sources TIGR RGD, IRGSP, Gramene, TAIR et BGI. Le schéma conceptuel s'inspire du schéma CHADO, proposé par le consortium GMOD. Nous avons complété et adapté ce schéma. (dont un extrait est reporté figure 4.3) qui intègre les caractéristiques des

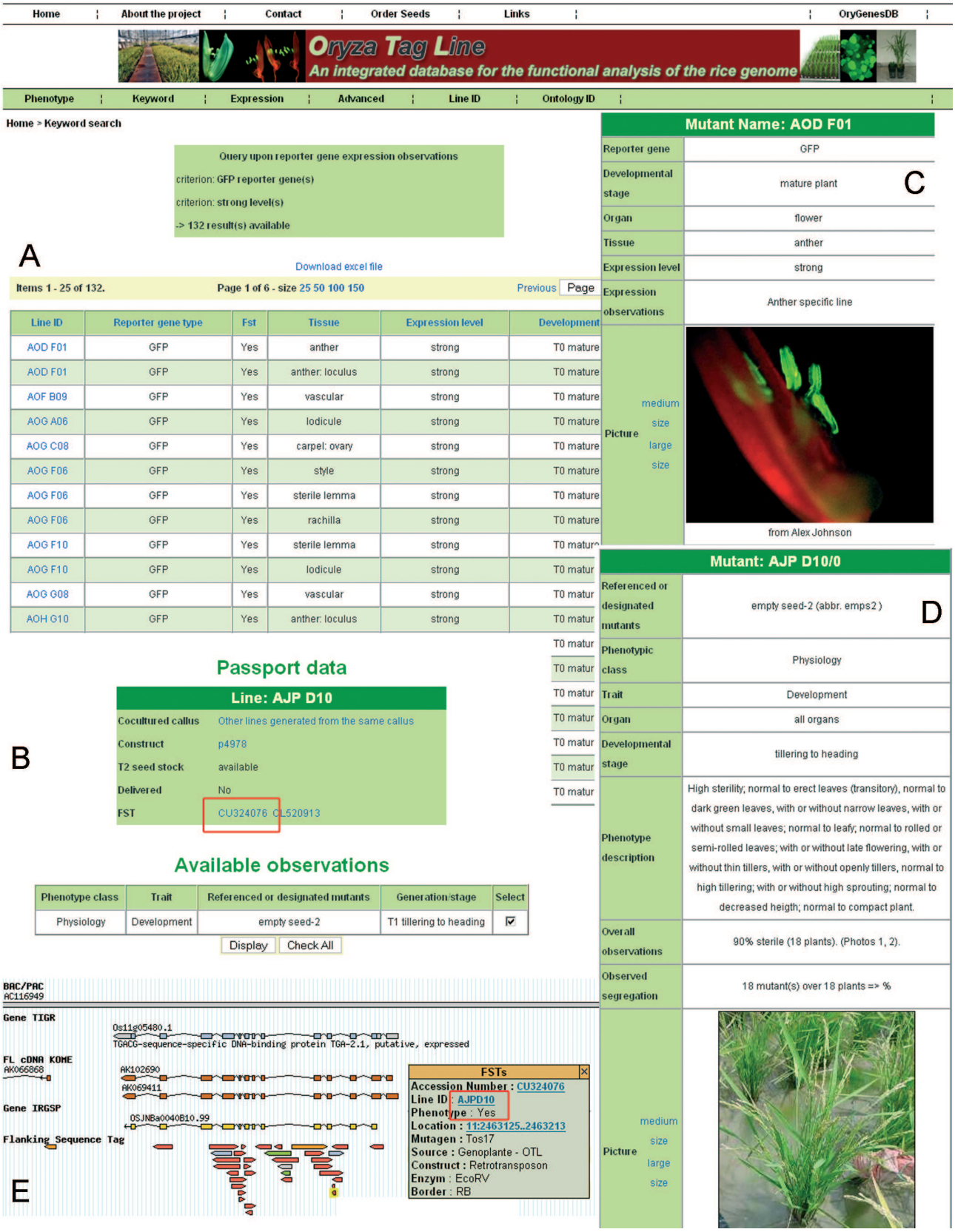


FIG. 4.2 – Description d'une recherche de mutant

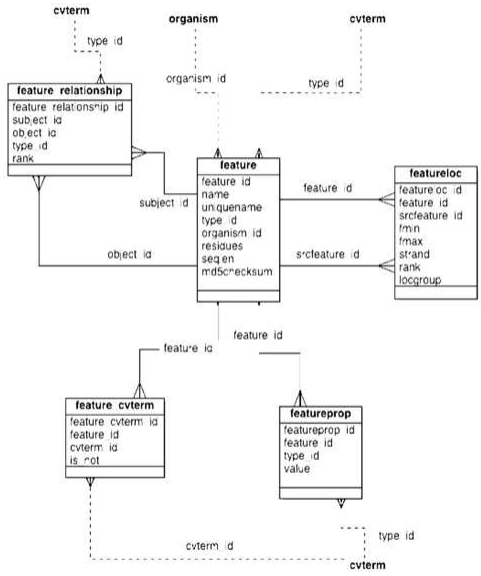


FIG. 4.3 – Description des principales tables du module séquence pour le modèle chado. Le module séquence est organisé autour de l'entité feature. Elle est générique et représente tous les éléments d'annotation d'une séquence qui peuvent interagir avec le système. (d'après Mungall et al. [MEC07])

différentes sources. Le cœur d'OryGenesDB est une base de données dont le socle est Mysql, les langages de programmation perl et perl-cgi.

4.3.1.2 Contenu

L'ensemble des données contenu provient de sources externes. L'insertion des données est réalisée par des programmes adaptateurs spécifiques à chacune des sources. Nous détaillons brièvement les modes d'alimentation de la base intégrée à partir des diverses sources. Le socle génomique de référence correspond aux 12 pseudo-chromosomes distribués par le TIGR[51]. Le génome et son annotation sont téléchargés via le site FTP du TIGR puis insérés dans la base. A cela est superposé la couche d'annotation "officielle" délivré par le consortium IRGSP[52] et disponible sur Genbank. Des programmes recalculent les nouvelles coordonnées des annotations en fonction du référentiel choisi. En effet, les pseudo-molécules correspondent à des assemblages de BAC, et sont donc dépendantes de l'état du séquençage ou des remaniements de ceux ci. Des versions de ces dernières sont régulièrement distribuées (une fois par an en moyenne). Nous intégrons toujours à partir du site FTP du TIGR les données Genes Indices, collection de clusters d'EST (Expressed Sequence Tags) spécifiques d'une espèce. Les clusters de plusieurs

[51] http ://www.tigr.org/tdb/e2k1/osa1/
[52] http ://rgp.dna.affrc.go.jp/IRGSP/

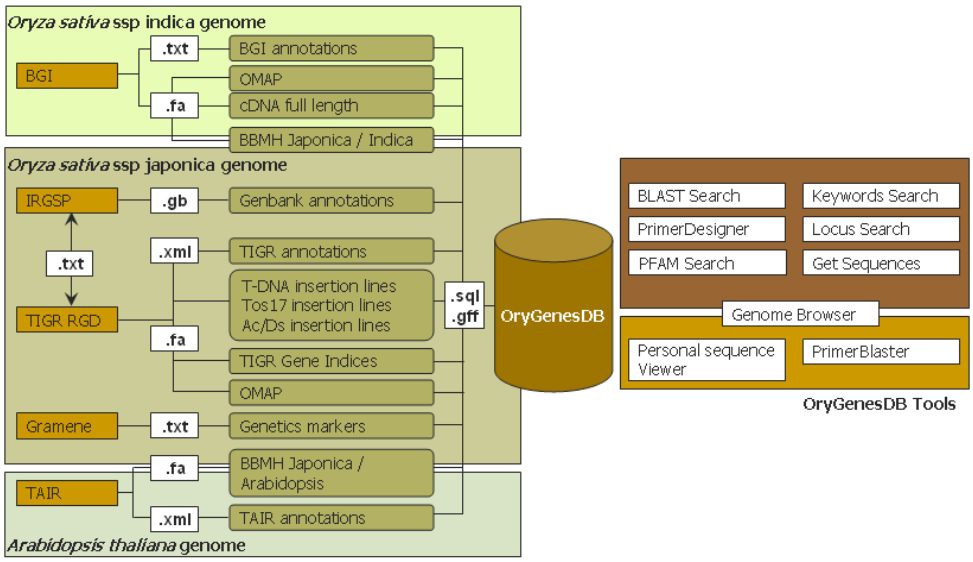


FIG. 4.4 – Description de l'origine des données dans OryGenesDB

Origin	Type	Number
Genoplante	T-DNA + Tos17	27,555 +, 1,348
Postech	T-DNA	80,006
CSIRO	T-DNA	787
RMD	T-DNA	15,727
TRIM	T-DNA	7,053
OSTID	Ds	1,380
PMBBRC	Ds	1,072
UCD	Ds	6,878
Genoscope	Tos17	13,017
NIAS	Tos17	18,024
Total		**172,847**

FIG. 4.5 – Description de la provenance des données FST

espèces ont été positionnés sur le génome du riz (blé, riz, maïs, etc.).

Les données relatives aux génomes d'*Oryza sativa* et d'*Arabidopsis* ont été chargées dans la base et nous avons conçu des traitements afin d'extraire les associations entre gènes de même fonction biologique. Par exemple, 10 679 paires de gènes orthologues ont été identifiées avec *Arabidopsis* en utilisant la méthode de BBMH (Best Blast Mutual Hit)[53].

Enfin des marqueurs génétiques provenant de la source Gramene ont été stockés dans la base. OryGenesDB contient des données de FST (T-DNA et Tos17) issues notre propre projet mais intègre également des données publiques provenant d'autres collections de mutants. La figure 4.5 décrit l'ensemble des données FST stockées dans la base ainsi que leurs provenances.

[53]BBMH : cette méthode consiste à exécuter des blasts sur les deux protéomes. Les gènes qui sont identifiés comme orthologues auront un alignement réciproque entre les deux espèces

En plus de l'intégration des données génomique, l'atout principal du système réside dans son interface graphique paramétrable de visualisation du génome qui est une adaptation du navigateur de génome (GBrowse) développé par le Generic Model Organism Project (GMOD).

Le système met de plus à disposition plusieurs outils d'analyses et de recherches approfondies intégrés et accessibles grâce à une interface Web. Dans ce domaine, OryGenesDB utilise de nombreux développements communautaires comme Bioperl pour le traitement de tâches spécifiques. Par exemple, pour parser des fichiers Genbank ou TIGR XML contenant des annotations ou traiter des résultats d'alignements (BLAST, BLAT, CLUSTAL, etc.).

4.3.2 Résultats

Nous avons déjà souligné précédemment qu'un atout du système réside dans l'interface principale d'OryGenesDB. L'interface est un navigateur de génome fonctionnel et très utile pour afficher des annotations sur un génome. Elle permet d'agrandir une portion de génome et de sélectionner les couches d'annotation que l'on désire afficher. Le navigateur a une fonction de recherche de type texte sur toutes les couches d'annotation disponibles. La figure 4.6 représente une portion de génome visualisé à travers GBrowse. A cette base graphique nous avons ajouté des fonctionnalités améliorant les performances de l'outil. Par exemple, au dessus du navigateur une représentation graphique des 12 chromosomes permet de naviguer plus rapidement en cliquant sur une portion du dessin. Une fenêtre s'affiche lorsque le pointeur de la souris passe sur les objets d'annotation. Cette fenêtre est utile pour afficher des informations qui ne sont pas présentes dans la description de l'objet. Ici les informations pour le gène 'Osg01010.2' permettent d'être redirigé vers la source d'origine (e.g. TIGR) et d'accéder également à des données stockées dans d'autre sources grâce aux liens de références croisées.

4.3.2.1 L'interface de requête

L'interface de requête est intégrée à l'interface principale et permet le lancement de divers types de requêtes.

Recherche par mots clés Lorsqu'une recherche par mots clés doit être lancée, les utilisateurs peuvent écrire plusieurs mots clés en utilisant des opérateurs (AND, OR et NOT). Une option de recherche par FST permet de sélectionner le type de FST (T-DNA, TOS17, etc.) alors que 'Gene Annotations' permet de rechercher de l'information dans les champs textes des données provenant des sources TIGR, IRGSP, cDNA ou de toutes. Une option 'region' permet de restreindre la recherche des FST en fonction de leur zone d'insertion (dans un promoteur, dans un gène, etc.) et l'option 'orientation' permet de choisir le sens de l'orientation. Tous les résultats (liste de gènes) sont affichés sous forme de tableau et exportables sous format Excel. Chaque élément du tableau possède un lien vers sa position graphique.

Recherche par domaine Cette interface se base sur les domaines PFAM [BCD+04] et Interpro [MAA+05] qui sont deux classifications de domaines protéiques conservés. Comme dans l'interface précédente il est possible de contraindre la recherche par type de FST, région ou orientation. Tous les résultats (liste de gènes) sont également affichés sous forme de tableau et exportables sous format Excel. Chaque élément du tableau possède un lien vers sa position graphique.

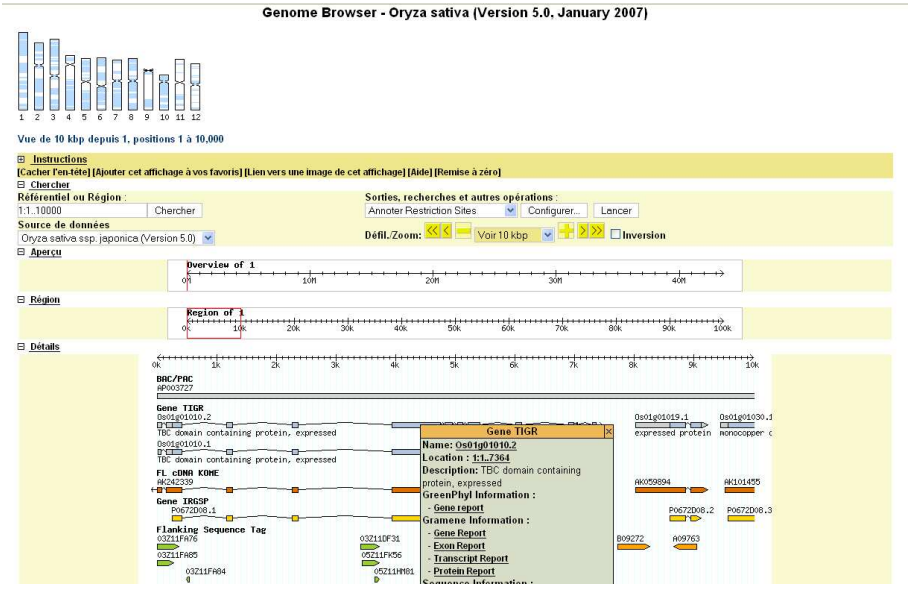


FIG. 4.6 – Illustration du navigateur GBrowse à travers OryGenesDB

Recherche par BLAST L'identification de FST dans les gènes peut être faite en exécutant le programme BLAST sur les séquences favorites des utilisateurs. Cette interface permet de soumettre une séquence qui est ensuite analysée pour trouver des FST. Les résultats sont affichés sous forme de tableau listant gène et FST à proximité de la séquence. L'outil permet également de visualiser directement le résultat sur le navigateur.

Ajouter des annotations sur le génome Cet outil permet de positionner une séquence (issue d'un autre site) graphiquement sur le génome du riz avec ses annotations. Il génère également un fichier GFF qui peut être réutilisé dans une autre application.

Recherche de locus L'interface de recherche de locus permet de soumettre une liste de noms de gènes afin d'obtenir les FST dont l'insertion est proche.

4.3.3 Discussion

OryGenesDB est un système d'information génomique développé pour la génétique inverse chez le riz. Il stocke préférentiellement les données FST produites dans notre laboratoire et intègre les données d'autres ressources publiques issues de projets similaires. La majorité des projets de génomique fonctionnelle utilisent des bases pour stocker leurs ressources mais elles ne sont pas toutes accessibles et ne possèdent que des interfaces de recherche basiques. Quelques unes comme TIGR, RiceGE, Flagdb++ et Gramene utilisent un navigateur de génome. OryGenesDB est actuellement la ressource qui possède la plus importante collection de FST (plus de 163 000). De plus, elle est avec RiceGe, la seule qui est spécialisée dans la génétique inverse chez le riz.

OryGenesDB présente de nombreux outils (programmes) qui permettent la simplification de la tâche des généticiens moléculaires. Souvent complémentaires de l'interface de navigation de génome (GBrowse) ils permettent par différents moyens de rechercher des insertions dans des gènes candidats.

4.4 Intérêt de l'intégration

Les deux premiers systèmes que nous avons proposés et développés ont une approche intégrative explicitée au moment de la conception.

D'autres systèmes ont été mis en place avec les mêmes objectifs comme GreenPhylDB qui a été développée dans notre équipe. Il s'agit d'une ressource de génomique comparative entre les deux génomes modèles (i.e. *Arabidopsis thaliana* et *Oryza sativa*).

Mais on ne peut en génomique fonctionnelle concevoir de système idéal centralisé avec un modèle global intégré, il faut donc mettre en œuvre l'intégration de sources diverses.

Nous allons illustrer ce besoin par un exemple. La figure 6.3 illustre les interactions entre OryGenesDB, OryzaTagLine et GreenPhylDB ainsi que l'information récupérée en dehors de ces systèmes.

Dans l'exemple illustré, la recherche débute avec le locus AT4G10380 du génome d'*Arabidopsis* sur l'interface TAIR apportant une première indication sur la fonction du gène. Dans le but de vérifier l'existence d'un gène ayant une fonction similaire chez le riz, la recherche se poursuit sur GreenPhylDB en utilisant AT4G10380 comme motif de recherche. Le résultat permet de mettre en évidence une relation simple entre AT4G10380 et un gène de riz. En effet, des gènes orthologues peuvent être trouvés avec ce système qui utilise des méthodes phylogénétiques pour identifier les relations entre les deux génomes. Le gène de riz équivalent Os10g36924.1 permet de d'obtenir de nombreuses informations dans la base OryGenesDB. Notamment, sa position sur le génome, son environnement génique et les différentes couches d'annotation. La couche «Flanking Sequence Tag» permet de vérifier si des mutants d'insertion ont été identifiés comme relié à ce gène. Dans ce cas, la séquence FST CL520431 indique qu'une plante (AIVB06) constitue un mutant de ce gène. Pour avoir plus de détails sur les observations phénotypiques effectuées, les biologistes utilisent un lien de référence croisée basé sur le nom de la plante. OryzaTagLine affiche alors un phénotype semi-nain avec des descriptions et des images associées.

Comme nous pouvons le constater, chacun des systèmes apporte de l'information qui est complémentaire des autres et qui permet de naviguer vers d'autres systèmes.

C'est pour arriver à reproduire de manière automatique et intégrée les recherches d'information à travers ces systèmes que nous nous sommes donc employé à réaliser deux approches différentes d'intégration l'une par médiation, l'autre par chorégraphie de services Web.

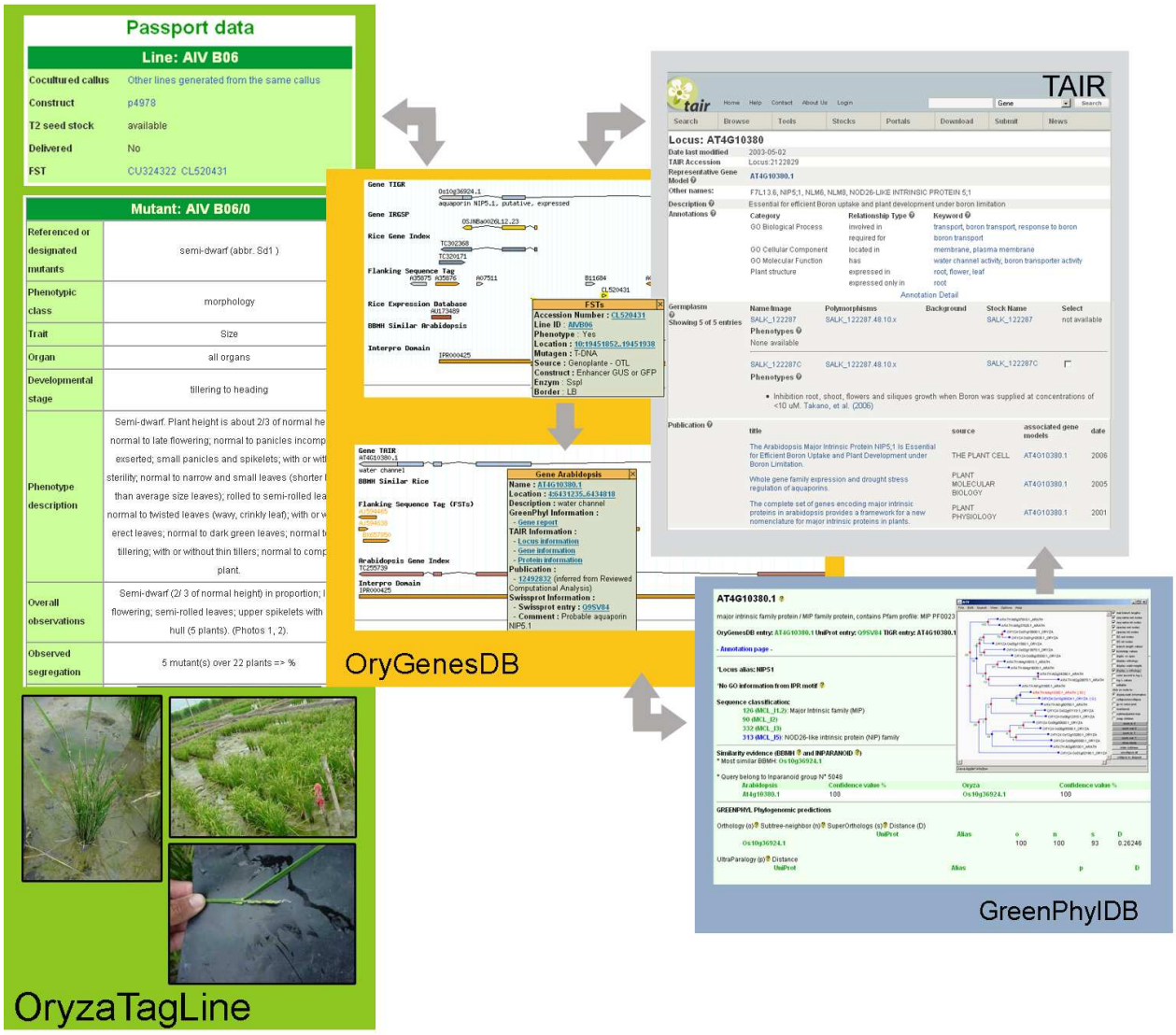


FIG. 4.7 – Navigation web au travers des sources OryGenesDB, Oryza Tag Line, GreenPhylDB et TAIR

Chapitre 5

Adaptation de Le Select pour la médiation de ressources végétales

Sommaire

Le riz est une denrée alimentaire de la plus haute importance, et se trouve à ce titre, une des céréales les plus cultivées au monde. Il possède également des qualités remarquables d'un point de vue génomique, qui le désignent tout naturellement comme plante modèle des monocotylédones. Son génome est en effet relativement petit par rapport à d'autres espèces végétales et présente des régions génomiques de forte similarité (conservation de la synténie) avec d'autres céréales (blé, orge, maïs, sorgho). Plus en avant, il possède une particularité qui lui confère un attrait supplémentaire. Le séquençage de son génome, achevé depuis Décembre 2004, révèle un nombre inattendu de gènes estimés entre 40000 et 60000. Pour comparaison, Arabidopsis thaliana, l'espèce modèle des dicotylédones en compte 27000. Une meilleure compréhension du "pourquoi" de la présence de ce répertoire de gènes hors du commun, passe par la détermination de la fonction biologique de l'ensemble de ces gènes ou encore par la confrontation de ces gènes avec les gènes provenant de génomes d'autres espèces végétales. Une approche de génétique inverse, plus précisément dite de mutagénèse insertionnelle, apporte des éléments de réponse précieux quand à la fonctionnalité ou encore au profil d'expression des gènes. La mutagénèse insertionnelle permet d'insérer, de manière aléatoire, dans le génome, un fragment d'ADN identifiable appelé ADN-T (ADN de transfert) ou élément transposable. Lorsque l'ADN-T s'intègre dans un gène, il peut en altérer la fonction et provoquer la modification du caractère observable chez l'individu, en correspondance (impact sur le phénotype). Le gène, ainsi muté, est localisé grâce à l'élément inséré et sa fonction est identifiée grâce au caractère affecté. De concert avec les projets internationaux qui ont pour objectif d'analyser, par mutagénèse insertionnelle, l'ensemble des gènes du riz, et plus précisément, dans le cadre du programme national de génomique végétale, Génoplante, le CIRAD s'est doté d'une collection de 30000 lignées de mutants d'insertion et de 40000 FST (extrémités flanquant le site d'insertion de l'ADN-T). Pour accompagner l'effort réalisé autour de la construction de lignées et pour en exploiter judicieusement toute l'information, le CIRAD a conçu et mis en place deux bases de données avec une visée intégrative à plus ou moins long terme :

- OryGenesDB qui regroupe l'ensemble des données génomiques publiques disponibles concernant le riz.
- Oryza Tag Line (OTL) qui contient des données phénotypiques sur une collection de mutant d'insertion T-DNA chez le riz.

Les informations présentes dans OryGenesDB et OTL sont complémentaires. Les rendre partageables, peut prendre tout son sens, dès lors qu'il s'agit de vouloir identifier rapidement l'effet d'une mutation dans un gène donné, en recherchant les impacts morphologiques ou physiologiques dans les plantes correspondantes. OryGenesDB et dans un moindre degré OTL, nécessitent une forte intégration des données mais aussi des traitements. Il est ainsi nécessaire de pouvoir consulter les séquences génétiques contenues dans OryGenesDB au moyen de systèmes de requêtages spécifiques basés par exemple sur de la recherche de similarité (au travers par exemple de l'outil BLAST) ou encore sur de la recherche de signatures de domaines protéiques fonctionnels (au travers de l'outil ScanProsite) OryGenesDB apparaît déjà comme une "mosaïque" d'informations intégrées provenant soit de projets nationaux et internationaux, soit

de sources de données internationales et autonomes. Dans OryGenesDB, se pose donc déjà le problème de maintenir et de faire évoluer de l'information en adéquation avec les évolutions réalisées au sein des sources de données dont elles sont originaires. OTL, s'appuie elle aussi sur Plant Ontology pour une meilleure description des traits phénotypiques, qui elle aussi, est sujette à évolution. Dans un souci d'ouverture, d'autres sources de données pourraient se révéler d'intérêt en terme de mutualisation, que cela soit des sources de données contenant des données chevauchantes (autres bases de données génomiques du riz par exemple) ou que cela soit des sources de données donnant accès à des informations complémentaires. C'est le cas par exemple d'une autre base de données du CIRAD, Greenphyl qui contient des données synthétiques sur la comparaison des répertoires géniques du riz et d'Arabidopsis thaliana. Il reste à l'esprit que ces différentes sources peuvent s'exprimer au travers de formats hétérogènes. Les besoins dégagés en termes de partage de données dans ce contexte, sont de différentes natures. Pour les résumer, il va donc s'agir de :

- faciliter l'accès aux données comme aux traitements (le chaînage des traitements doit également pouvoir être envisagé).
- laisser une totale autonomie et indépendance aux sources de données sous-jacentes. Le système de médiation ne doit en aucun cas prendre le contrôle et la maîtrise des données.
- le nombre de sources de données prises en charge par le système de médiation doit pouvoir évoluer en fonction des besoins des végétalistes. En outre l'hétérogénéité des formats des sources de données ne doit pas se révéler un obstacle.
- le partage des sources de données tout comme leur consultation doivent être facilités. En outre, il doit être possible de poser des requêtes complexes au système, pouvant porter sur plusieurs sources de données distantes sans que les temps d'attente pour les réponses, ne deviennent prohibitifs.

Un premier objectif est, ici, d'exploiter une architecture de médiation permettant à des communautés d'utilisateurs (ici une communauté scientifique de végétalistes) de partager des ressources distribuées (données, programmes, et ressources de calcul). Il s'agit également au delà du partage, d'intégrer les données afin de permettre l'expression de requêtes complexes portant sur des sources de données potentiellement hétérogènes. Traditionnellement, un schéma global de médiation, qui spécifie les correspondances sémantiques des sources de données, est alors défini. Les limitations portent alors sur le peu de flexibilité du schéma qui est maintenu de manière centralisée et qui ne va donc pas pouvoir prendre en charge les besoins spécifiques de chacun des membres de la communauté. Il va alors s'agir, dans un second objectif, de s'appuyer sur une architecture décentralisée afin que chaque usager puisse alors définir ses propres correspondances sémantiques ou exploiter différents schémas de médiation proposés au sein de la communauté. Le système de médiation Le Select, développé dès son origine, pour les besoins propres des communautés scientifiques, est mis à contribution afin d'apporter les éléments de réponse nécessaires à ces objectifs.

Le chapitre comprend cinq sections. Dans la première section, nous revenons sur les fondements du système de médiation Le Select. Une seconde section fait état des sources de données à mutualiser dans le contexte de notre travail. Les sections trois et quatre illustrent la démarche suivie et les résultats obtenus (adaptation de Le Select au contexte de l'étude). La section cinq vient clore le chapitre en proposant une synthèse sur les avancées et en explorant de nouvelles pistes.

5.1 Description du middleware

5.1.1 Principales caractéristiques

Le Select est un système middleware développé depuis 1998 par le projet caravel[54]dans le cadre du projet européen Thetis[55] pour répondre aux besoins des applications scientifiques de partager des données et des programmes. Le Select [MBFS02, Sim01, CMC+02, MBFS03] est un successeur du système Disco (développé dans le projet Rodin de 1995 à 1998 dans le cadre de l'action Dyade Médiation, puis transféré en 1999 à la société LibertyMarket qui commercialise le portail Kelkoo.com) [TRV98]. Le Select permet l'intégration et le partage de sources données hétérogènes distribuées avec un accès uniforme et intégrés à travers un langage de haut niveau. Les données ont une représentation uniforme exprimée dans le modèle de données relationnel étendu à des types de données définis par l'utilisateur (e.g. structurés, semi-structurés, etc). Écrit en Java, le médiateur propose également un accès uniforme à l'exécution de programmes intégrés (e.g. services, programmes) ainsi que la publication et le traitement des données issues de ces processus. De manière générale, Le Select offre des outils de transformation des données publiées et permet d'attacher une documentation structurée sur ces dernières.

D'un point de vue réseau, Le Select possède une architecture distribuée de type mediateur/adaptateur, ce qui veut dire qu'il n'existe pas de dépôt centralisé pour intégrer les données, ni de schéma global. En effet plusieurs applications Le Select peuvent coopérer pour fournir l'accès aux ressources. Publier par exemple, des systèmes d'information par l'intermédiaire de Le Select évite de mettre à jour les données intégrées et maintient leur autonomie vis à vis d'autres applications clientes. Ces avantages, nous voulions les mettre à profit dans le cadre d'un projet scientifique visant l'intégration de ressources de données végétales. Nous détaillerons ce projet dans la section 5.2.

La figure 5.1 représente le déploiement d'un système utilisant Le Select. Deux grandes parties sont visibles, la partie "clients" et la partie "sites de publication". La partie "sites de publication" est propre à chaque serveur, elle permet l'intégration, la transformation et la publication des données et programmes. Par convention la publication de données est le fait de mettre les données à disposition sur un serveur Le Select. La partie "clients" communique avec les serveurs à travers le réseau. Les clients peuvent être des applications web utilisant des librairies Le Select spécifiques ou des navigateurs web. Les serveurs ont la possibilité de communiquer entre-eux pour répondre à une requête lancée par un client.

5.1.2 L'accès aux données

Le Select est un système middleware qui permet la publication de données et l'accès aux données publiées par des programmes. Le modèle de données utilisé dans Le Select est le modèle relationnel avec, pour exploiter les sources, un langage pivot proche du standard SQL. Le fait que Le Select utilise le standard SQL, lui permet d'interagir à un bon nombre d'applications s'appuyant sur ce standard.

[54]Le projet Caravel (http ://www-caravel.inria.fr), transformé depuis en projet SMIS (Systèmes de Médiation d'Information Sécurisé), avait pour thème fondateur l'intégration de sources d'information avec à la fois la facilitation de la publication de ressources ainsi que la facilitation de la production de données.

[55]Le Projet Européen Thetis porte sur la gestion des ressources côtières

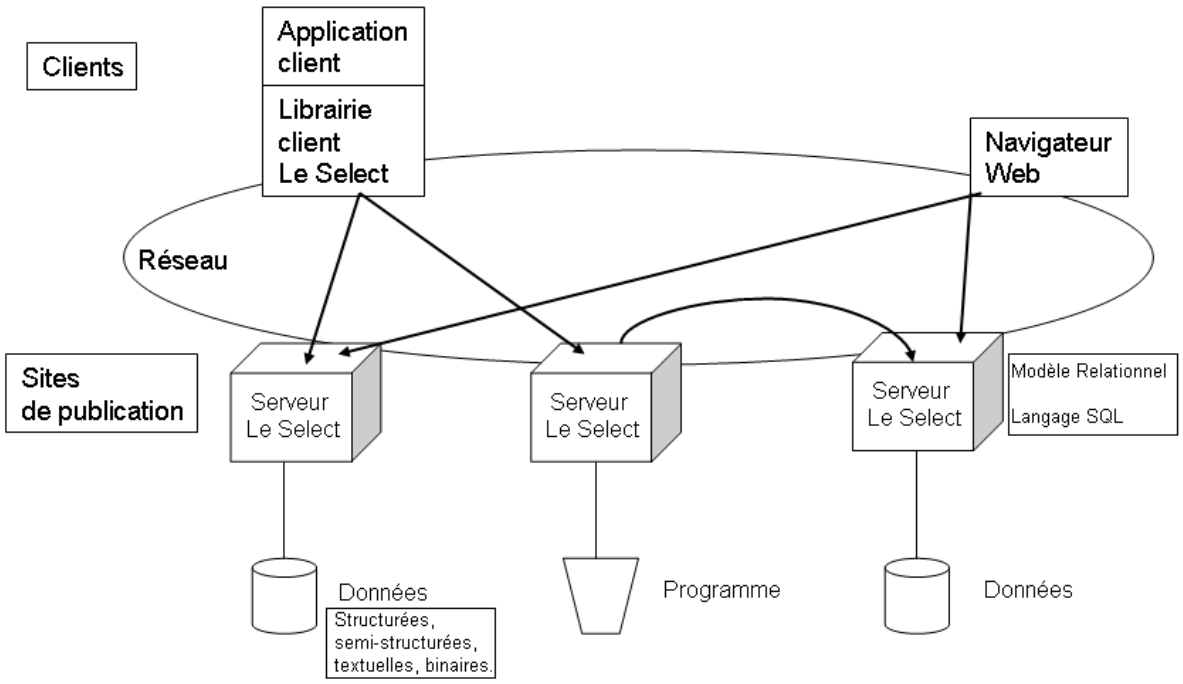


FIG. 5.1 – Architecture de Le Select

5.1.2.1 Le rôle des adaptateurs

La publication des données, qu'elles soient stockées dans des fichiers plats, des bases de données ou générées à la volée par un programme, est gérée par un adaptateur ou *wrapper* en anglais. De manière générale, le rôle des *wrappers* est multiple. (i) Ils jouent un rôle de traducteur entre le médiateur et la source en transformant les données de leur modèle d'origine vers le modèle pivot (e.g. modèle relationnel dans le cas de Le Select). (ii) Ils exportent des méta-données sur les ressources qu'ils publient telles que la disponibilité des ressources, le temps d'exécution des requêtes, etc. Ils exportent également des informations sur leurs capacités d'exécution de requêtes (e.g. jointure, union). Toutes ces méta-données servent au médiateur pour optimiser les requêtes à travers un plan d'exécution. Dans le cas de Le Select les wrappers exportent de la documentation sur la resource et les données qu'elle contient.

Les adaptateurs de données Pour chaque source de données devant être publiée, un *wrapper* existant doit être utilisé ou créé *de novo* si les *wrappers* communs ne conviennent pas. Le Select contient 3 types de *wrapper* : texte délimité, texte tabulé et JDBC. Le premier est utilisé pour publier des données dans des fichiers plats ASCII, dans lesquels la structure est organisée en ligne avec des valeurs séparées par un caractère délimitant (i.e. les fichiers csv). Le *wrapper* tabulé a les mêmes caractéristiques avec une valeur fixe de délimitation. Le wrapper JDBC permet d'accéder à n'importe quelle base de données via JDBC. La ré-utilisation de ces *wrappers* se fait par l'intermédiaire d'un fichier de configuration de type XML, *wrapper definition file*. Les *wrappers* utilisent ces fichiers pour accéder aux sources (e.g. paramètres de connexion, déclaration des types et noms des attributs de colonnes, etc), par conséquent un fichier de configuration est nécessaire pour chaque source publiée.

```
AEBG11.jpg;/picture/plant/Osmu2/large/AEBG11.jpg;image/jpg;raw
AEBG11_01.jpg;/picture/plant/Osmu2/large/AEBG11_01.jpg;image/jpg;raw
AGZB12_01.JPG;/picture/plant/Osmu2/large/AGZB12_01.JPG;image/jpg;raw
AGZB12_02.JPG;/picture/plant/Osmu2/large/AGZB12_02.JPG;image/jpg;raw
AING04_01.JPG;/picture/plant/Osmu2/large/AING04_01.JPG;image/jpg;raw
AING04_02.JPG;/picture/plant/Osmu2/large/AING04_02.JPG;image/jpg;raw
```

FIG. 5.2 – Exemple de fichier de données portant sur les images de plantes

La figure 5.2 représente les données contenues dans une source de type texte (i.e. picture.txt). Elle même listant l'ensemble des images disponibles pour la collection de plantes du CIRAD. La figure 5.3 représente un exemple de ces fichiers pour une source de type texte. Le type de *wrapper* est défini par l'attribut "WrapperClass" dans la première ligne (1) du document en paramètre de l'élément "Wrapper". Cet attribut est obligatoire et désigne le nom de la classe Java correspondant au type de wrapper. L'élément "Wrapper" contient deux éléments "Parameters" et "Documents". Le premier, obligatoire, définit les éléments qui vont être publiés alors que le deuxième, facultatif, sert à attacher de la documentation aux *wrappers* et aux tables. La structure de l'élément "Parameters" varie en fonction du *wrapper* utilisé. Dans ce cas, il possède l'attribut "separator" permettant de distinguer les différents élément d'une ligne et contient au moins un élément "Table" (4). Le bloc "Table" (4 à 9) correspond aux éléments de la source picture.txt publiée. L'élément "Table" contient un attribut "name" ayant comme valeur le chemin relatif vers le fichier texte. Il contient également deux éléments nommés "Column" (5 à 8). Les éléments ont des attributs "name" et "type" correspondant respectivement aux noms et types que prennent ces colonnes. Les informations incluses entre les balises documents servent à documenter le wrapper sous la forme de méta-données (voir la section 5.1.2.1).

```
1 <Wrapper WrapperClass="LeSelect.Wrappers.Text.TextWrapperFactory">
2
3   <Parameters separator=";">
4    <Table name="Picture" file="data/picture.txt">
5            <Column name="Nom" type="string" />
6            <Column name="Chemin" type="string" />
7            <Column name="TypeMime" type="string" />
8            <Column name="Bin" type="raw" />
9    </Table>
10  </Parameters>
11  <Documents>
12       ...
13  </Documents>
14 </Wrapper>
```

FIG. 5.3 – Exemple de fichier de configuration de wrapper texte

L'interrogation des données issues de ressources de type "fichier plats" s'effectue de la même manière qu'une requête SQL. Par exemple si nous souhaitons filtrer la liste d'images en effectuant une projection sur la colonne nom :

(Q1) : Trouver les noms, chemins et binaires des images dont le début du nom commence par AGZB12.

La requête se traduira dans Le Select de cette manière :

```
select Nom, Chemin, Bin from /picture where nom like 'AGZB12%'
```

FIG. 5.4 – Exemple de requête Le Select sur une source de données

Les adaptateurs de programmes Le Select possède également les fonctions nécessaires à la publication de programmes. Un *wrapper* de programme convertit les données d'entrées en tables de manière équivalente aux données publiées et dans un format convenable pour le programme, puis exécute le programme à partir d'une requête SQL. La particularité du middleware est de permettre l'exécution de programmes de manière distribuée (par exemple, un programme local sur données distantes). L'exécution de programme est traitée de manière asynchrone, palliant ainsi à un délai de réponse trop long. Un identifiant d'exécution est alors généré par le *wrapper*, il s'agit du $jobID dans la figure 5.6. Il permet alors de vérifier l'état d'exécution du programme (par exemple, en attente, erreur, terminé...). Les données sont retournées sous forme de tables au même titre que les données publiées. Pour que cela soit possible, un *wrapper* de données est utilisé. En effet, après que le *wrapper* de programme termine l'exécution, le médiateur lui demande un wrapper de données pour accéder aux résultats. Les *wrappers* de programmes possèdent également d'autres fonctions. Par exemple, ils permettent d'informer le médiateur du coût d'exécution du programme, de la gestion des accès concurents, et de la gestion des résultats.

La figure 5.5 représente le fichier de configuration pour le wrapper de programme (wdf) adapté à la recherche de similarité de séquences au travers du programme BLAST. Le fichier de configuration indique dans sa première ligne le type de wrapper utilisé "GenericXMLPrFactory". L'élément "Parameters" contient 3 sous éléments qui correspondent aux commandes ("ProgPath") et fichiers de transformation pour les données d'entrées ("XSLInPath") et sorties ("XSLOutPath"). Les informations saisies dans la balise "Documents" permettent de documenter le wrapper de programme.

```
<ProgramWrapper
WrapperClass="programwrapper.generic.GenericXMLPrWrFactory">
  <Parameters>
    <ProgPath value="java BlastCommand"/>
    <XSLInPath value="wrapperconf/BlastProgram_in.xsl"/>
    <XSLOutPath value="wrapperconf/BlastProgram_out.xsl"/>
  </Parameters>
</ProgramWrapper>
```

FIG. 5.5 – Exemple de fichier wdf pour un programme BLAST

(Q2) : Exécuter le programme BLAST en utilisant les séquences de la table temporaire fasta

Dans l'exemple d'exécution d'un programme BLAST (figure 5.6), nous pouvons détailler trois instructions typiques ; le lancement du programme effectué par la commande EXECUTE. Dans ce cas c'est le wrapper de programme qui est appelé avec les paramètres nécessaires à son exécution ainsi que les données d'entrée ici extraites d'une table temporaire (input dataset is select * from /temp/fasta). Le deuxième ordre consiste à vérifier l'état d'exécution du programme avec la commande QUERY. Si le résultat n'est pas terminé le même identifiant est retourné sinon le chemin de la table résultat est renvoyé. La dernière instruction permet la consultation des résultats stockés dans une table temporaire.

```
JOB EXECUTE /programs/blast
   parameter prog = 'blastn'
   parameter db = rice_genome
   parameter evalue = 5
   input dataset is select * from /temp/fasta
JOB QUERY $jobID
SELECT * FROM $tmpTableName
```

FIG. 5.6 – Exemple de requête Le Select sur un programme BLAST

Les mécanismes de vues Le Select possède des fonctionnalités qui lui permettent d'effectuer des transformations sur les données publiées. C'est un mécanisme de vues. Elles permettent d'effectuer des transformations sur les données ou d'intégrer des données issues de différentes sources. Par exemple, dans la vue définie en figure 5.8 nous intégrons les noms de plantes provenant de deux sources sous le même nom d'attribut (i.e. name). Comme nous pouvons le voir sur la figure, elles se construisent de la même manière que des requêtes SQL ordinaires et peuvent bénéficier de l'utilisation des opérateurs ensemblistes comme l'opérateur d'UNION qui va permettre de faire l'union entre deux relations de même schéma. Une fois définies, elles peuvent être utilisées par des clients comme s'il s'agissait d'une table sans distinction avec les données publiées.

```
<view>
   <definition query=
            "select a.name as name from /bcrddb/material a
            union
            select b.NAME_LINE as name from /oryzatagline/line b" />
</view>
```

FIG. 5.7 – Exemple de vue Le Select sur l'intégration de données issues de sources différentes

L'attachement de méta-données Le middleware possède des outils de documentation des ressources publiées. Par exemple des informations sur l'unité de mesure d'une valeur ou l'auteur de la donnée et sa date de création peuvent être stockées dans un document spécifique. Ce document est mis en ligne de la même manière que les données elles même servant alors de base pour des recherches par un moteur de recherche. La documentation peut décrire les *wrappers* ou les tables d'un *wrapper* (un *wrapper* peut définir plusieurs tables). Il peut être inclus dans le

fichier de configuration du *wrapper* ou inséré dans un autre fichier dont le lien est indiqué dans le fichier de configuration.

FIG. 5.8 – Exemple de documentation du *wrapper* texte picture

```
<Documents>
   <WrapperDocument>
        <attribute name="picture" value="Source listant toutes les images
 de plantes disponibles dans la collection de mutant d'insertion" />
   </WrapperDocuments>
   <TableDocument tableName="picture">
        <attribute name="Nom" value="Nom de l'image correspondant au mutant">
        <attribute name="Chemin" value="Chemin de la localisation des images">
        <attribute name="TypeMime" value="TypeMime des images">
        <attribute name="Bin" value="Accès aux binaires des images">
   </TableDocument>
</Documents>
```

L'élément "Documents" comprend les parties nécessaires pour décrire les *wrappers* (WrapperDocument) et les tables (TableDocument) associées aux *wrappers*. Un élément "WrapperDocument" possède obligatoirement au moins un élément "attribute" qui est lui-même décrit par une paire "name" et "value". Dans ce cas, l'attribut "name" correspond aux noms des sources publiées par le *wrapper*. Dans le cas de l'element TableDocument l'attribut "name" correspond aux noms des colonnes publiées.

5.1.2.2 Le rôle du médiateur

La principale fonction du médiateur Le Select est de fournir un accès homogène aux sources de données. Son architecture est divisée en plusieurs composants :

- L'architecture centrale
- Les *wrappers*
- Les modules d'authentification
- Le serveur
- Les modules de communication

L'architecture centrale implémente les principales caractéristiques de Le Select. Elle permet aux données d'être publiées en utilisant les *wrappers* de données, exécute les requêtes utilisateurs, garde trace des documents attachés aux données, se charge d'exécute puis de publier les vues. Par ailleurs, elle permet la publication des programmes via un *wrapper* approprié et en gère l'invocation.
L'interface de cette architecture utilise le driver JDBC. On peut voir son comportement comme celui d'une base de données ; les tables sont construites à partir des *wrappers* et les requêtes sont exécutées en soumettant des sous-requêtes aux *wrappers* correspondants. Les programmes sont également invoqués à travers JDBC grâce à un langage d'invocation spécialement spécifié et implémenté.
Dans cette section nous allons détailler plus précisément l'architecture centrale de Le Select et nous parlerons des modules de communication.

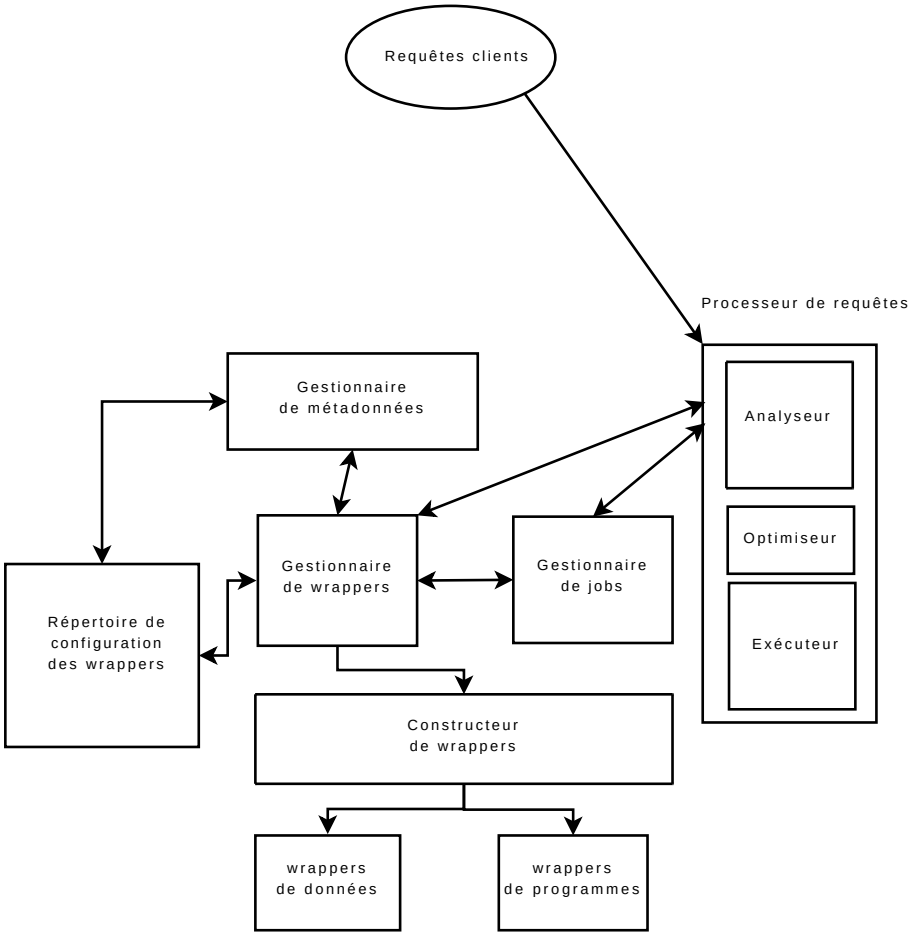


FIG. 5.9 – Architecture Centrale simplifiée de Le Select

L'architecture centrale de Le Select La figure 5.9 représente l'architecture centrale de Le Select, avec de manière simplifiée ses principaux composants. Tout d'abord au niveau de la publication, le répertoire de configuration des wrappers contient les informations nécessaires pour donner accès aux sources de données. Les fichiers de types XML vont servir à implémenter des informations au niveau du gestionnaire de wrapper et du gestionnaire de métadonnées. Comme nous avons pu le voir précédemment ce sont les fournisseurs de sources qui interagissent avec cette partie. Le gestionnaire de wrappers est l'élément important du dispositif puisqu'il interagit avec de nombreuses autres parties. Par exemple il permet la création des différents wrappers en passant par son constructeur (c'est à dire un constructeur Java), il récupère des métadonnées pour les fournir à l'analyseur de requêtes et l'informe sur l'exécution des programmes. Le gestionnaire de job s'occupe de fournir les données d'entrée aux programmes, de démarrer leurs exécutions et fournir les résultats.
Lorsqu'une requête est effectuée par une interface client, un traitement est décomposé en plusieurs étapes par le "Query Engine". L'analyseur va sélectionner les sources potentiellement capables de répondre à la requête. L'optimiseur va sélectionner les sources en tenant compte des informations que lui renvoie le gestionnaire de wrapper afin d'établir un plan d'exécution de requêtes et diviser la requête principale en sous-requêtes. L'exécuteur effectuera le traitement des requêtes, en les dirigeant vers les wrappers appropriés ou lancera une exécution de programme si nécessaire.

Les modules de communication Une fois les données publiées, elles sont accessibles par des interfaces clients. Le moyen le plus utilisé est par le biais de l'interface JDBC, mais les données sont accessibles également via les protocoles FTP et HTTP car Le Select les implémente. Il propose notamment une interface de consultation et de requête par le biais du serveur HTTP (voir figure 5.12)

5.2 Description de l'intégration des sources

5.2.1 Description des sources

En génomique fonctionnelle, rassembler des informations provenant de domaines complémentaires permet de conforter les hypothèses émises et de découvrir de nouvelles relations entre les objets biologiques. Nous allons décrire des ressources qui jouent un rôle important dans les processus d'analyses biologiques. Comme nous l'avons décrit dans le chapitre 1, les ponts établis entre données génomiques (OryGenesdb) et phénotypiques (Oryza Tag Line) sont essentiels pour la compréhension du fonctionnement des gènes. De plus, les informations apportées par d'autres sources (Rice-BRCdb) ajoutent une validation supplémentaire à l'analyse. Dans ce sens, l'objectif de notre approche est de rendre interopérable des bases de données hébergées par différents instituts (CIRAD - IRD - CINES), afin de permettre leur utilisation de manière transparente. La figure 5.10 représente l'organisation de ses ressources.

OryGenesDB est une base de données (MySQL - CINES) intégrative pour les ressources génomique du riz[56] [DRL+06]. En plus de contenir toutes les informations du génome et de ses annotations, elle stocke des informations génomiques (FST) sur la collection de plantes mutantes. Ces étiquettes FST permettent de faire un lien entre le gène muté et le mutant observé pour son phénotype. Grâce à ce lien, les données de OryGenesDB et Oryza Tag Line sont exploitables pour des analyses transversales.

Oryza Tag Line (OTL) est une base de données (Oracle - CIRAD) phénotypique développée sous Oracle (CIRAD)[57]. Elle contient, les données d'observations phénotypiques de la collection Génoplante de mutant de riz.

Rice-BRCdb est une base de données (MySQL - IRD) sur les ressources génétiques riz développées et disponibles en France. Elle contient des données phénotypique et génétiques des différentes collections de riz produites par le CIRAD et l'IRD.

En plus des bases de données cité précédemment, d'autres ressources sont nécessaires comme les images des plantes mutantes de la collection de riz Génoplante et le programme BLAST utile pour rechercher une séquence d'ADN d'intérêt dans le génome du riz.

5.2.2 Publication des sources

Afin de publier les resources à partir du médiateur, des *wrappers* spécifiques ont été créés. Par exemple, un *wrapper* JDBC a été réutilisé pour effectuer les différentes connexions aux bases de données. Pour chaque base de données un fichier de configuration comme celui de la figure 5.11 a été créé. Le fichier fait référence à une DTD spécifique des wrappers JDBC et

[56] http ://orygenesdb.cirad.fr/

[57] http ://urgi.versailles.inra.fr/OryzaTagLine : est l'url publique de la base mais une instance privée est hébergée au CIRAD

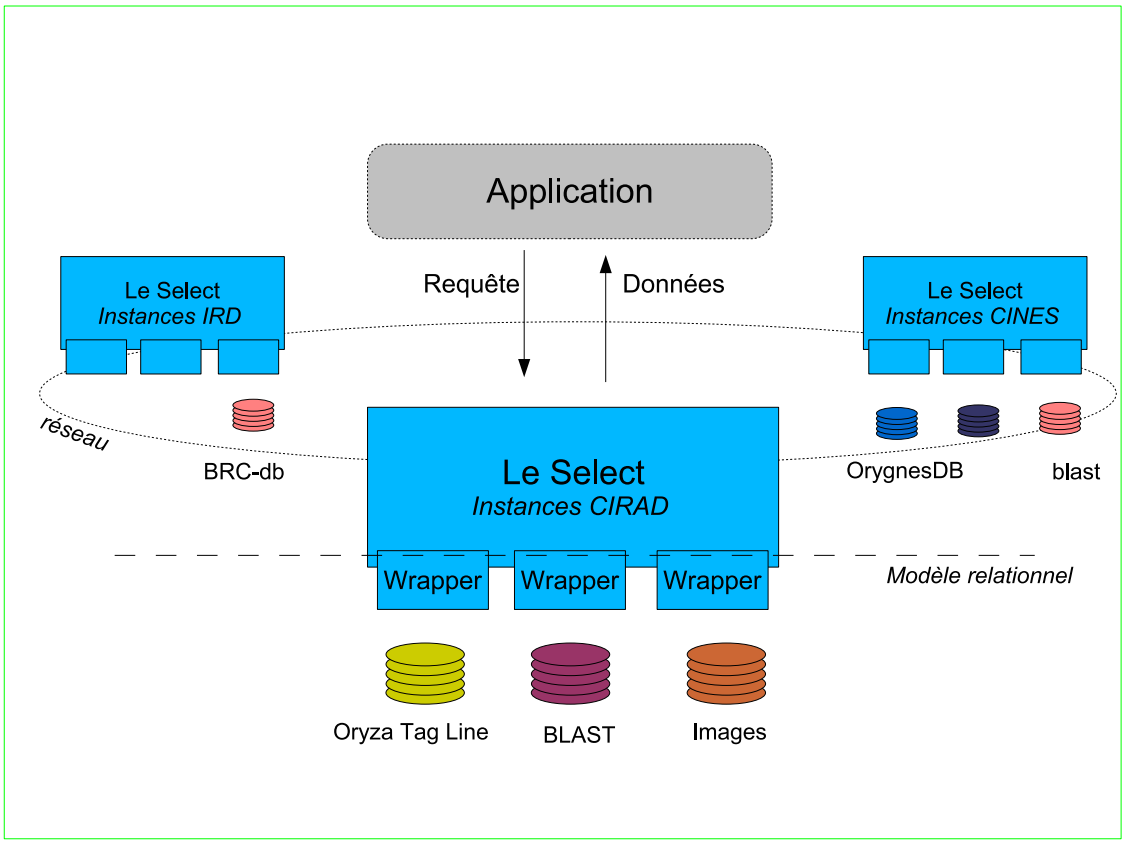


FIG. 5.10 – Organisation des sources de données publiées par Le Select

```
 <?xml version="1.0" encoding="UTF-8" ?> <!DOCTYPE Wrapper
SYSTEM "leselect:dtd/jdbc_wrapper.dtd"> <Wrapper
WrapperClass="LeSelect.Wrappers.Jdbc.JdbcWrapperFactory">
        <Parameters
                jdbcClass="org.gjt.mm.mysql.Driver"
                url="jdbc:mysql://valois.cirad.fr:3306/brcdb"
                user=""
                password=""
                quotedNames="no"
        >
        </Parameters>
</Wrapper>
```

FIG. 5.11 – Exemple de fichier de configuration de wrapper texte

utilise le type de "WrapperClass" équivalent. Ensuite le document est composé d'un élément "Parameters" lui même possédant 5 attributs. Parmi ces derniers nous retrouvons les paramètres classique d'une connexion JDBC à savoir le type de driver, l'url de connexion, le user ainsi que le mot de passe.

La figure 5.12 montre comment s'articule la publication d'OTL via Le Select. Sur cette page nous distinguons plusieurs parties. La partie de gauche liste toutes les sources disponibles à travers les *wrappers* créés, alors que la partie du haut permet de changer de vues (e.g. *wrapper* , *view*, *program*). Dans la partie centrale, s'affichent les données sous forme de table. Elles peuvent correspondre au résultat d'une requête effectuée dans la zone de texte en bas de l'interface ou au contenu d'une table listée dans la zone des sources. Dans l'exemple présenté sur cette figure, les données proviennent de la table TRAIT et représentent les caractères phénotypiques observés dans la collection de mutants de riz. Les données sont directement extraites de la base de données par le *wrapper*, dans ce cas la structure de la table n'est pas modifiée.

5.3 Intégration sémantique des sources de données

Nous allons respecter, d'un point de vue chronologique, les activités, qui de manière classique, constituent le processus d'intégration dans le monde des bases de données. Dans ce sens, les activités de pré-intégration des schémas initiaux, de recherche de correspondances au sein de ces schémas et puis au final d'intégration de ces schémas locaux sont conduites [PS96]

5.3.1 Pré-intégration

Nous prenons en entrée les schémas locaux des bases de données OryGenesDB, OTL et BRC-DB. Les schémas pour ce qui concerne OTL et BRC-DB sont partiellement chevauchants. Le schéma d'OryGenesDB est complémentaire. Il s'agit, lors de cette première phase, de faciliter les étapes de recherche de correspondances inter-schémas et d'intégration à venir, en uniformisant tant que faire ce peu, les schémas considérés en entrée. Les schémas conceptuels d'OryGenesDB, d'OTL et de BRC-DB sont exprimés tout trois sous la forme de diagrammes de classes UML. Ces schémas ont été également traduits sous la forme de schémas relationnels. D'un point de vue syntaxique, les modèles sont donc déjà uniformisés. Il est rappelé que le

FIG. 5.12 – Oryza Tag Line publiée par Le Select

modèle relationnel se trouve également être le modèle de données pivot proposé par Le Select. D'un point de vue sémantique, il existe un certain non-déterminisme de la modélisation puisque les modèles en entrée ont tous été définis par des concepteurs différents dans des institutions différentes (CIRAD et IRD). De plus, les contextes dans lesquels les sources de données ont été construites sont quelque peu différents. Ainsi, la source OTL est centrée exclusivement sur la représentation d'observations phénotypiques sur des lignées de mutants et uniquement chez le riz. BRC-DD possède un focus d'intérêt plus large et modélise des données phénotypiques provenant de lignées mutantes comme de lignées sauvages et ce, chez différentes espèces de plantes. Enfin, OryGenesDB s'attache à modéliser les concepts biologiques exploités en génomique fonctionnelle.

Nous décrivons, ici et de manière très précise, les concepts d'importance afin de pouvoir au mieux identifier les correspondances par la suite.

Dans le cadre d'OTL (figure 5.13), une lignée de mutants *Line* est produite à partir d'une construction *Construction* et peut être vue comme une composition de plantes mutantes *Plant*. La généalogie des plants mutants est conservée au travers d'une association réflexive de filiation *is_parent_of* de *Plant* vers *Plant*. Chaque plant mutant est décrit au travers d'observations phénotypiques *Eval_pheno* à différentes étapes de développement. Une observation phénotypique *Eval_pheno* est, en outre, à un trait biologique, très généralement de nature qualitative *Trait*. Les traits biologiques sont enrichis par des termes *Ontology_element* provenant des vocabulaires contrôlés (e.g. Plant Ontology, Plant Growth stage Ontology,Trait Ontology and Plant structure Ontology [IKJ+07, PJK+06, YJ05, WJN+02, JAI+05]). Pour chaque observation phénotypique *Eval_pheno* ou construction *Construction*, des images peuvent être associées *Picture*. Dans le cadre de BRC-DB (figure 5.14), une lignée *Line* peut être soit une lignée sauvage, soit

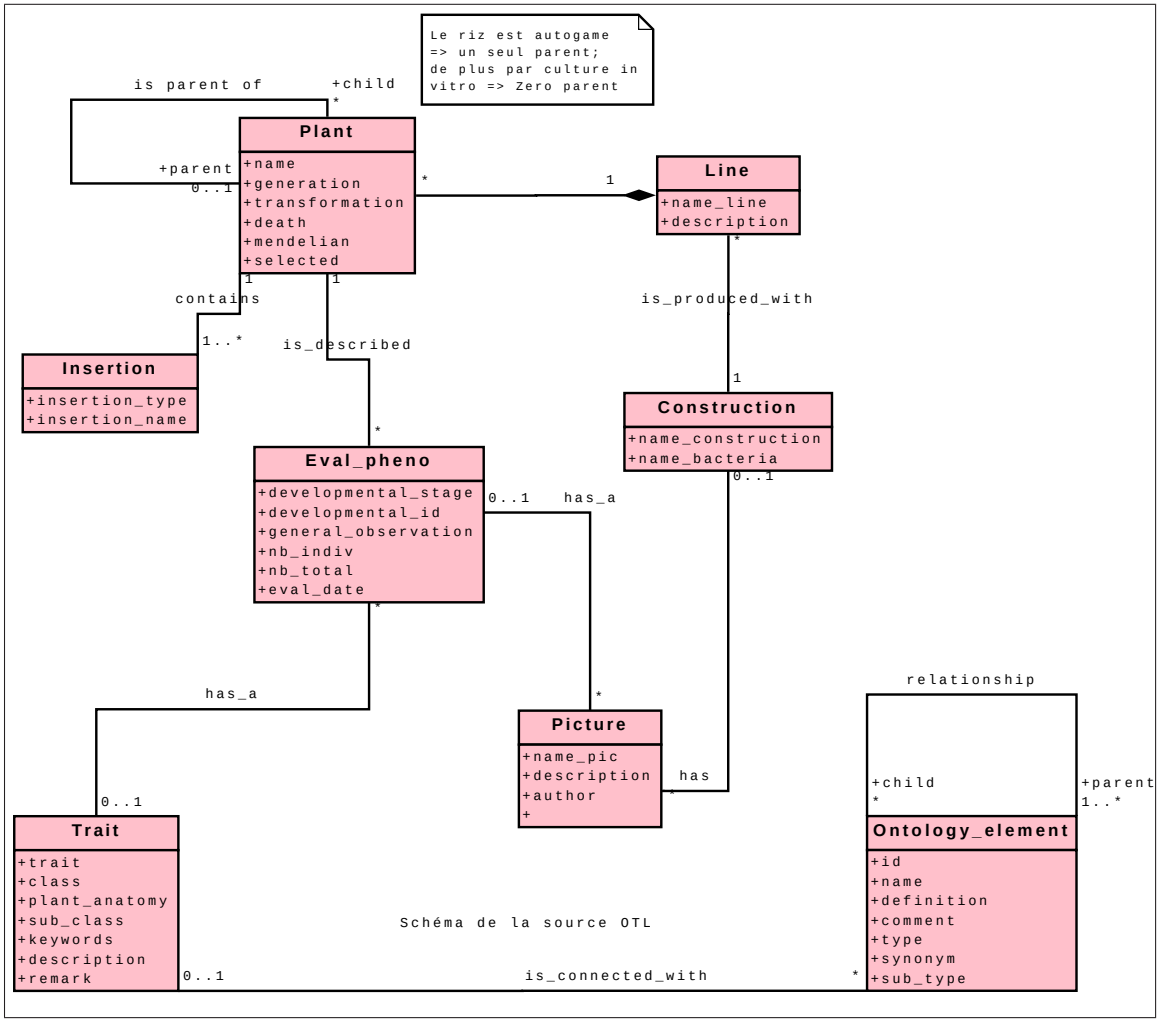


FIG. 5.13 – Illustration d'une partie du schéma OTL

une lignée mutante. Elle est vue comme une composition de plantes sauvages ou mutantes *Plant*. La classe matériel biologique *Material* généralise les classes *Plant* et *Line*. Une relation de filiation est modélisée au travers d'une association réflexive de *Material* vers *Material*. *Material* est attaché à un taxon *Taxon*. Les différents niveaux taxonomiques sont également représentés au travers d'une association réflexive de *Taxon* vers *Taxon*. Chaque plante *Plant* est décrite au moyen d'observations phénotypiques *Phenotypic_observation*. La description d'une observation phénotypique *Phenotypic_observation* est enrichie par des traits biologiques, représentés ici par des termes empruntés *Ontology_element* au vocabulaire contrôlé Gramene. Les termes du vocabulaire (hyperonymie, hyponymie) sont raffinés au travers d'une association réflexive d'*Ontology_element* vers *Ontology_element*.
Dans le cadre d'OryGenesDB (figure 5.15), l'élément central est l'unité de transcription représenté par la classe *Transcript_unit* d'où sont produit les gènes identifiés par la classe *Gene_model*. Un autre type de ressources biologiques sont les FST (Flanking Sequence Tag, voir chapitre 1.3.2.1). Ils sont reliés aux gènes par *FST_in_gene*. La classe *Mapping_FST* représente des informations de localisation des FST sur le génome. *Sequence* concerne les informations sur les FST elles mêmes, elle est liée à *Sequence_alias* qui symbolise les différents noms que peut prendre les séquences. Enfin, *FST_ressources* rassemble des données non biologique sur les

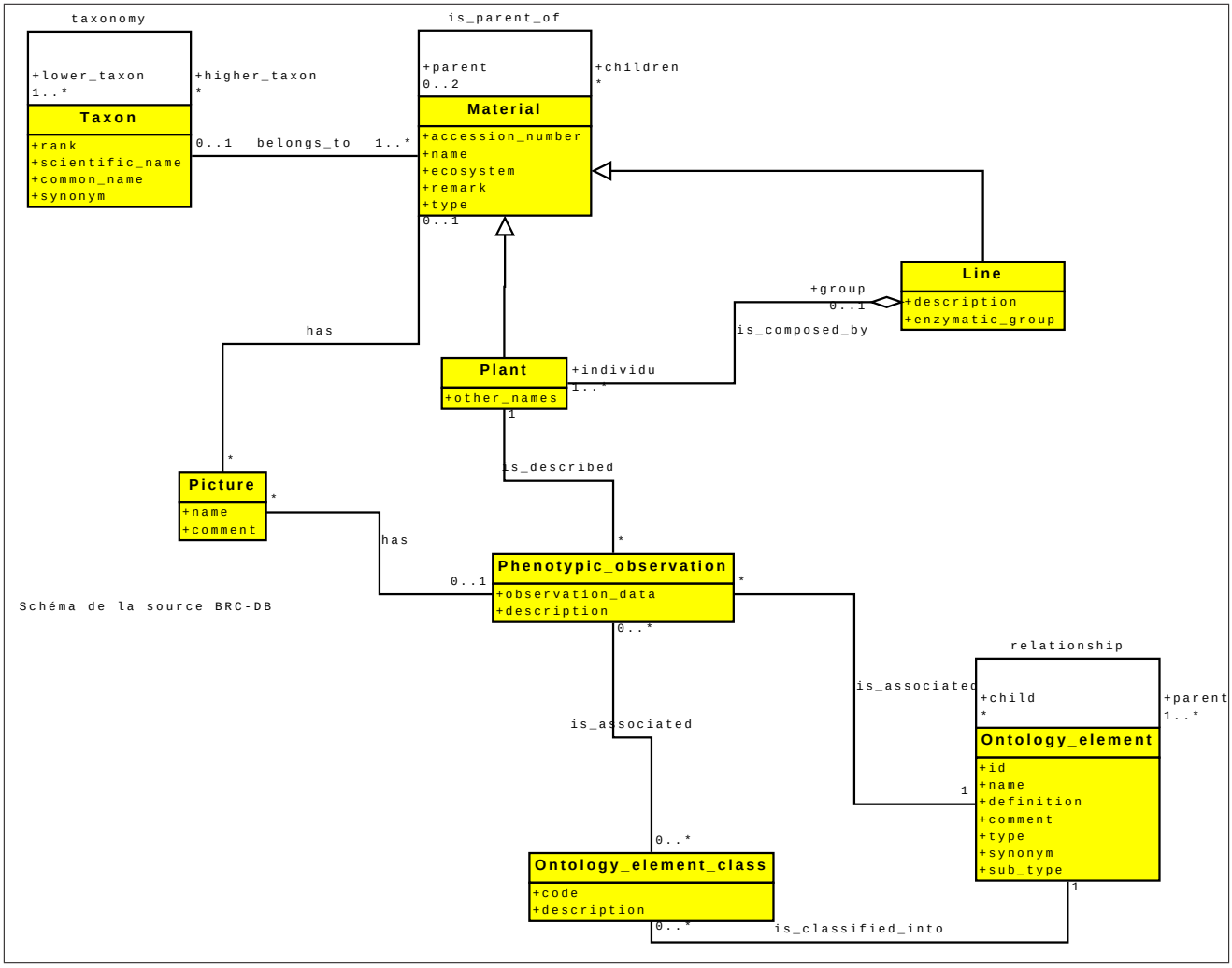


FIG. 5.14 – Illustration d'une partie du schéma BRC-DB

FST.

5.3.2 Recherche de correspondances inter-schémas

Nous allons maintenant, à partir des classes identifiées comme étant partagées par les trois schémas, définir des correspondances en intention. Ces correspondances sont nommées des assertions de correspondance inter-schémas (ACI) et vont servir notamment à dégager les classes non matérialisées du schéma global qu'il nous reste à construire. Nous exprimons les ACI au travers d'expressions algébriques s'appuyant sur les opérateurs de la théorie des ensembles et sur les opérateurs spécifiques de l'algèbre relationnelle. De manière générale, deux situations vont se présenter :

1. les classes partagées sont identiques ; nous nous trouvons dans le contexte d'une réplication de classes ou encore de mapping identité
2. les classes partagées présentent des différences de représentation ; nous nous trouvons dans le contexte d'un conflit de fragmentation qui peut être précisé sous forme de conflit de classification, de description ou encore de structure.

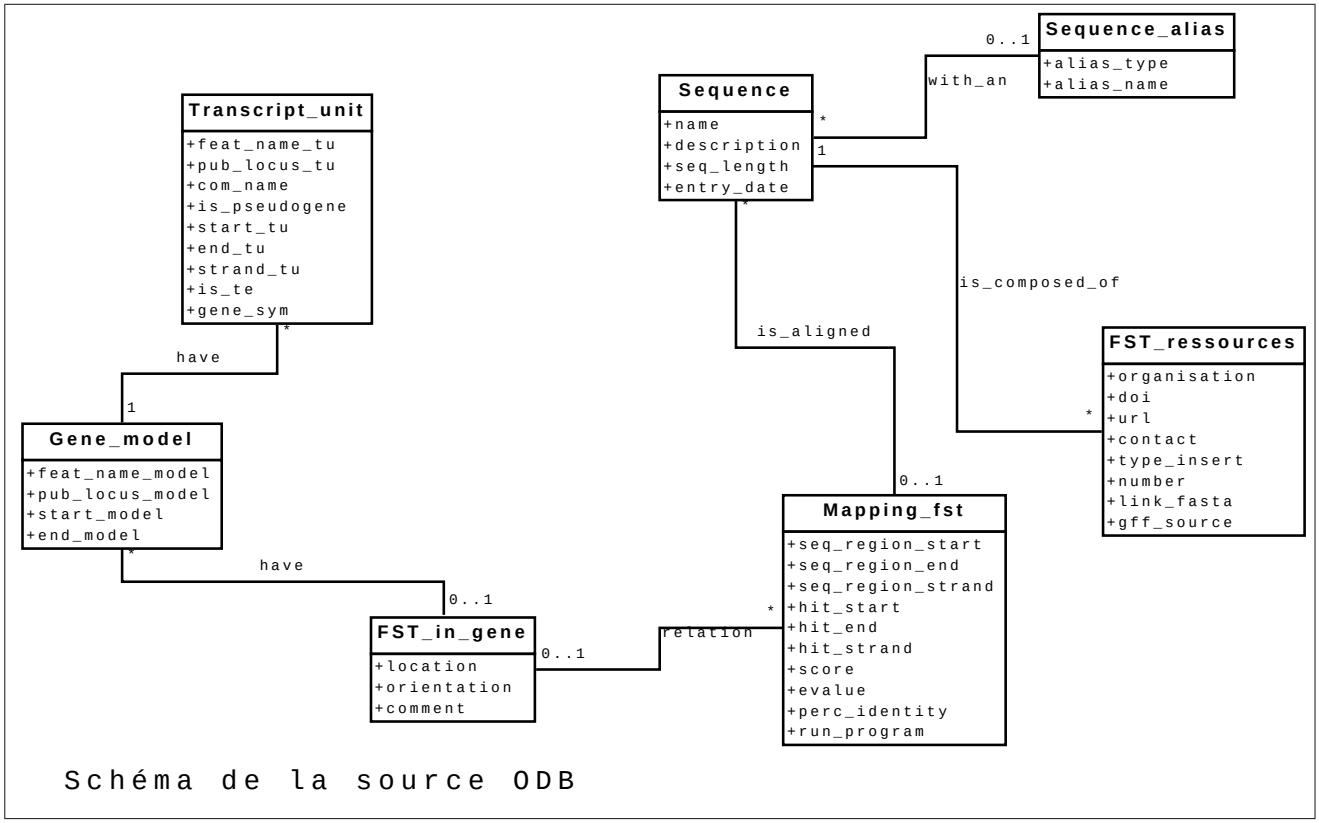


FIG. 5.15 – Illustration d'une partie du schéma OryGenesDB

Nous enrichissons, de plus, les ACI en donnant les attributs en commun des classes en correspondance (Avec Attributs Correspondants (AAC)).
Les conflits de fragmentation, dégagés ici, sont essentiellement des conflits de classification et des conflits descriptifs. Les concepts *Plant* et *Line* d'OTL et de BRC-DB représentent des ensembles d'instances fortement connectés mais différents puisque OTL ne modélise que les plantes et les lignes mutantes dans le seul cadre du riz alors que BRC-DB modélise les plantes et les lignées à la fois sauvages et mutantes chez différentes céréales.
Il nous est alors possible de décrire ce conflit par l'une des deux ACI ci-dessous :

$$OTL.Plant \subseteq BRC-DB.Plant$$

$$\sigma_{[\text{est une plante mutante}]} BRC-DB.Plant \equiv OTL.Plant$$

Nous retiendrons l'écriture de l'expression algébrique s'appuyant sur l'opérateur de sélection, plus riche à nous yeux d'un point de vue sémantique.
Material est un concept très général décrivant tout type de matériel biologique qui généralise *Line* comme *Plant*. De fait, différents concepts n'auront pas d'équivalents dans les schémas locaux. Ainsi le concept *Taxon* classifiant les espèces végétales associées aux plantes et lignées n'est présent que dans BRC-DB (inutile dans OTL puisque seul le riz est considéré) et le concept *Construction* n'est présent que dans OTL (BRC-DB plus généraliste, ne prend pas en charge les processus de construction des lignées mutantes). OTL et BRC-DB intègrent toutes deux les vocabulaires contrôlés proposés par Gramene pour ajouter du sens et qualifier les observations phénotypiques réalisées. L'intégration n'a cependant pas été réalisée de la

même manière Ainsi dans OTL, un concept *Trait* vient en complément des observations phénotypiques et *Trait* est qualifié ensuite par le concept *Ontology_element* décrivant les éléments des vocabulaires contrôlés. Pour ce qui concerne BRC-DB, les observations phénotypiques sont directement qualifiées par le concept *Ontology_element* décrivant les éléments des vocabulaires contrôlés.
D'un point de vue des conflits descriptifs, les ensembles d'attributs caractérisant les classes sont souvent différents. Ainsi *Eval_pheno* est caractérisé par un ensemble d'attributs que l'on ne va pas forcément retrouver dans *Phenotypic-observation*. Des attributs de classes en correspondance vont cependant être des synonymes ; c'est le cas par exemple de OTL.Line.name_line et de BRC-DB.Line.name.
Enfin les attributs en correspondance peuvent être définis au travers de types de données différents. C'est le cas par exemple, pour les attributs eval_date et observation_date définis respectivement au travers des types chaîne de caractères et date. Dans ce contexte, il suffit d'appliquer une fonction de conversion adaptée pour aller vers l'unification.

Ensemble des ACIs

Nous proposons ci-dessous l'ensemble des assertions de correspondance inter-schémas dans son expression minimale.
L'ojectif est par la suite de s'appuyer sur cet ensemble pour en dériver un schéma global, des règles de correspondance afin de construire des vues dans le médiateur Le Select et enfin un composant ontologique. Nous enrichissons , de plus, les ACI en donnant les attributs en commun des classes en correspondance (Avec Attributs Correspondants (AAC)).

$\sigma_{[\text{est une plante mutante}]} BRC-DB.Plant \equiv OTL.Plant$

$\sigma_{[\text{est une lignee mutante}]} BRC-DB.Line \equiv OTL.Line$

$OTL.Eval\_pheno \equiv BRC-DB.Phenotypic-observation$
avec AAC OTL.general_observation = BRC-DB.description

$OTL.Picture \equiv BRC-DB.Picture$
avec AAC OTL.name_pic = BRC-DB.name
et AAC OTL.comment = BRC-DB.description

$OTL.Trait \bigcap BRC-DB.Ontology\_Element$

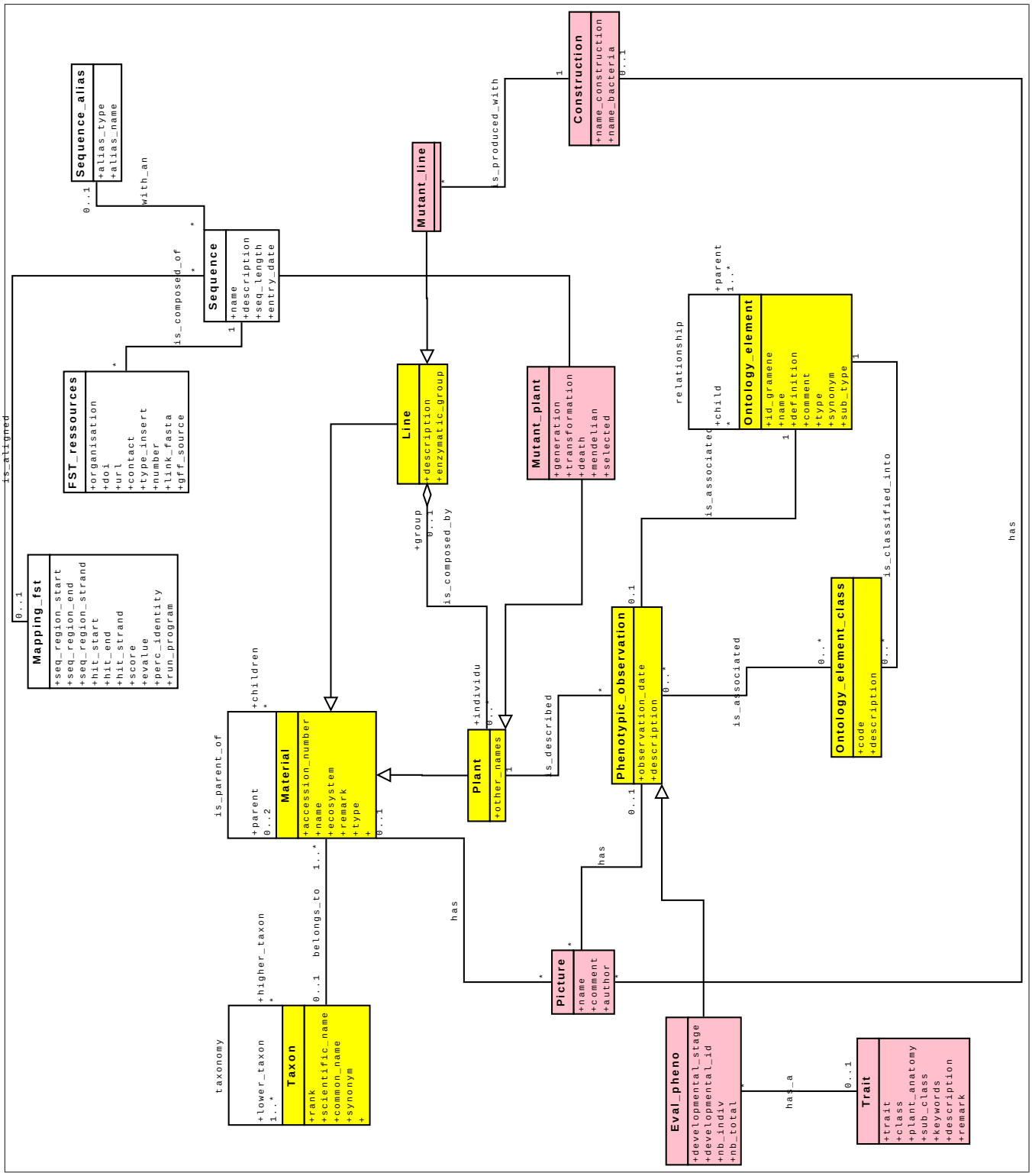
$OTL.Ontology\_Element \equiv BRC-DB.Ontology\_Element$

$\sigma_{[\text{est une construction}]} BRC-DB.Material \equiv OTL.Construction$

$OTL.Insertion \equiv ODB.Sequence$
avec AAC OTL.insertion_name = ODB.name

5.3.3 Intégration

Deux approches s'entendent dès lors que l'on veut définir un schéma intégré. Il est en effet d'usage de privilégier soit la simplicité et ainsi de rendre minimal l'ensemble des concepts dégagés au niveau du schéma intégré ; soit la l'exhaustivité et la complétude de la représentation unifiée. Nous avons utilisé principalement cette dernière approche en y incluant des hiérarchies de généralisation/spécialisation dans la résolution des conflits. Ce schéma est présenté dans la figure 5.16.


Sequence_alias
+alias_type
+alias_name
with_an
Sequence
+name
+description
+seq_length
+entry_date
is_composed_of
Mutant_line
is_produced_with
Construction
+name_construction
+name_bacteria
FST_ressources
+organisation
+doi
+url
+contact
+type_insert
+number
+link_fasta
+gff_source
Line
+description
+enzymatic_group
Mutant_plant
+generation
+transformation
+death
+mendelian
+selected
Ontology_element
+id_gramene
+name
+definition
+comment
+type
+synonym
+sub_type
relationship
+child
+parent
is_associated
Mapping_fst
+seq_region_start
+seq_region_end
+seq_region_strand
+hit_start
+hit_end
+hit_strand
+score
+evalue
+perc_identity
+run_program
is_aligned
+group
is_composed_by
+individu
Material
+accession_number
+name
+ecosystem
+remark
+type
is_parent_of
+children
+parent
Plant
+other_names
is_described
Phenotypic_observation
+observation_date
+description
is_associated
Ontology_element_class
+code
+description
is_classified_into
has
Picture
+name
+comment
+author
has
Taxon
+rank
+scientific_name
+common_name
+synonym
belongs_to
+higher_taxon
taxonomy
+lower_taxon
Eval_pheno
+developmental_stage
+developmental_id
+nb_indiv
+nb_total
has_a
Trait
+trait
+class
+plant_anatomy
+sub_class
+keywords
+description
+remark

5.3.4 Construction d'une ontologie

L'objectif est de réaliser une ontologie de domaine dans le but de partager un modèle commun pour les sources de génomique fonctionnelle végétales. En s'appuyant sur une ontologie, l'intégration de nouvelles sources est facilité [MSTH05]. Par exemple, l'utilisation de languages logiques comme les logiques de descriptions permet grâce aux mécanismes de classification et d'inférence, de mettre en correspondance les schémas locaux au schéma global sous forme d'ontologie. De plus l'évolution des systèmes utilisant une ontologie comme schéma global est facilité par ces mêmes mécanismes qui permettent d'ajouter facilement des nouveaux termes ou bien d'en supprimer. Dans certains cas, l'ontologie permet de définir des requêtes utilisateurs réalisées sur le schéma global [SGB00, SGHB02]. Enfin, l'ontologie peut se révéler une aide pour définir des vues ou réécrire des requêtes au niveau du médiateur.

Nous avons choisi de représenter des exemples extraits du schéma global (figure 5.16) avec le langage OWL DL (voir section 2.3.2 pour les logiques de descriptions). La traduction a été réalisée selon des règles établies par le document de spécification ODM [OMG07]. De manière générale, les classes UML trouvent leurs équivalents en OWL. Les relations de généralisation/spécialisation sont représentées par l'élément <rdfs :subClassOf>. Les attributs des classes comme les relations binaires sont représentées par l'élément <owl :ObjectProperty>, leurs cardinalités sont également valuées grâce aux restrictions <owl :Cardinality>, <owl :minCardinality>, <owl :maxCardinality>.

Pour l'exemple illustré figure 5.17, nous avons choisi de représenter les classes Material, Plant et Line ainsi que leurs relations. Il s'agit d'une partie importante du système d'intégration dans la mesure où les autres classes s'y rattachent afin d'enrichir les connaissances sur les plantes. De plus, les relations de parenté entre individus renforcent cette intégration.

Pour l'exemple de la figure 5.18, nous avons représenté les classes ayant un rôle dans les observations phénotypiques des plantes. Les classes Ontology_element et Ontology_element_class ont pour finalité de stocker des termes phénotypiques et des valeurs servant à décrire les plantes.

Dans le dernier exemple, figure 5.19 représente la relation entre l'identification d'une mutation dans un gène symbolisé par les classes, sequence, FST_in_gene et sequence_alias, et les plantes mutantes représentées par la classe Mutant_plant.

5.4 Interrogation transparente des sources

5.4.1 Construction des vues

Dans cette section, nous prenons comme exemple le modèle de la figure 5.16 pour réaliser les transformations sur le schéma des sources. Dans Le Select, les transformations s'effectuent par l'intermédiaire de vues sur les sources publiées (voir figure 5.20). Dans la définition des vues, nous avons tenu compte des identifiants de clé primaires et étrangères présents au niveau du modèle physique des sources, mais que nous n'avons pas représenté dans le diagramme de classe. Ces identifiants maintiennent la cohérence des données issues des différentes sources.

Construction de material : la vue material est composée d'éléments issus de 3 tables (BRC.material, OTL.plant et OTL.LINE). Le résultat de la vue (figure 5.20) correspond à l'union de ses 3 tables. Les attributs communs (par exemple, name) sont généralisés alors que les attributs absents du schéma local sont remplacés par des valeurs "Null".

Construction de line : comme la vue précédente, celle-ci est composée de l'union des deux tables (BRC.line et OTL.line). Seul l'attribut *description* est commun aux deux schémas locaux.

Construction de mutant_line : cette vue correspond à une spécialisation de OTL.line. Elle contient l'attribut *plant_number* spécifique à OTL.line. les identifiants présents dans le modèle physique sont ajoutées. Ils sont nécessaires pour maintenir certaines relation entre les entités.

Construction de mutant_plant : cette vue est construite essentiellement avec les attributs provenant de OTL.plant, c'est à dire generation, transformation, death, mendelian et selected.

Construction de picture : elle résulte également de l'union des deux tables picture dans les schémas locaux. Les vues permettent de filtrer les données par exemple pour ajouter un niveau de confidentialité. Ce qui est le cas ici, puisque les images prisent le photographe "perez" ne pas sont publiées à ce niveau.

Le Select possède de nombreuses fonctions utilisables pour transformer les sources publiées. Il est possible par exemple de convertir des types sur des attributs (e.g. toDate(eval_date)). Les transformations peuvent être réalisées sur les données grâce à des opérateurs de condition et des fonctions de manipulation de chaînes. Par exemple l'attribut material.type prend plusieurs valeurs en fonction du type de matériel étudié (e.g. seed, plant, line, germplasm, DNA, etc). Les valeurs germplasm et line sont synonymes, nous avons donc utilisé des opérateurs pour remplacer les valeurs provenant d'une source. Dans l'exemple ci-dessous, nous utilisons l'opérateur ifElse qui comprend 3 paramètres : la condition, la valeur de remplacement et la valeur dans le cas contraire.

```
ifElse(type = 'germplasm', 'line', toString(NULL)) as type
```

5.4.2 Exemples de requêtes

Afin de réaliser et valider les transformations à appliquer sur les sources, nous avons définis des requêtes mettant en jeu différents cas d'utilisation des sources.
(Q1) : Retourner la liste des noms de plantes ayant des photos prises par l'auteur 'mlorieux'
(Q2) : Retourner la liste des noms de plantes dont des phénotypes ont été observés pour le gene ERECTA
(Q3) : Retourner la liste des noms de plantes ayant des observations phénotypiques pour le caractère "biotic stress" et des photos

La requête Q1 est effectuée sur le schéma global et met en jeu la source Oryza Tag Line.

```
Q1: select m.name
    from material m, plant p, picture i
    where m.id=p.id and i.author='mlorieux' and i.material_id=m.id
```

La deuxième requête met en jeu des bases complémentaires OryGenesDB et Oryza Tag Line. Cela signifie que leur recouvrement est minime. Dans ce cas, il s'effectue sur une seule

correspondance, à savoir le nom des plantes. Un plan d'exécution peut être effectué pour traiter cette requête. Il est décomposé dans la figure 5.22. Les relations décrites sont adaptées du schéma global de la figure 5.16.

Finalement la requête Q3 est un exemple complexe de requête sur des sources chevauchantes. Elle utilise des vues (figure 5.20 et 5.21) définies entre les deux sources Oryza Tag Line et BRC-DB.

5.5 Conclusion

Dans ce chapitre nous avons montré que la médiation de sources de données de génomique fonctionnelle chez le riz était réalisable. L'adaptation de Le Select pour la publication des sources Oryza Tag Line, Brc-DB, Orygenes et Greenphyl permet de garder une autonomie vis à vis des sources tout en ayant un cadre uniforme de représentation et d'interrogation (i.e. accès uniforme à l'information). De ce fait la mise à jour des données au niveau des sources est automatiquement disponible au niveau du médiateur. De plus les moyens de transformation que présente Le Select permettent d'avoir une flexibilité dans l'ajout de nouvelles sources ce qui correspond aux attentes des biologistes. Si cette solution est toutefois contraignante puisque elle entraîne une modification du schéma global lors de l'ajout de sources, la contrainte est diminuée si les sources nécessaires sont complémentaires et donc ont un *mapping* peut important. Parmi ces moyens, le mécanisme de vues proposé par Le Select permet de résoudre les problèmes d'ordre syntaxique et sémantique en créant des correspondances entre les sources. Il a l'avantage de faciliter la réécriture des requêtes faites sur le schéma global.

Cette approche n'est toutefois pas entièrement satisfaisante. Une des contraintes est que les sources doivent être disponibles et accessibles. En effet, pour ajouter une source il faut y avoir accès (i.e. accès en lecture d'un fichier ou d'un répertoire, connexion à une base de données). Les problèmes de disponibilité des sources interviennent lors de l'exécution d'une requête, dans les cas où une connexion à une source est perdue. Mais l'évolution du schéma des sources pose également des problèmes de disponibilité sur l'exécution de la requête globale.

Lors de la publication des sources, l'étape de correspondance vers le schéma global peut s'avérer longue et contraignante à réaliser manuellement. Elle nécessite une grande connaissance des sources que l'on veut publier et peut prendre beaucoup de temps pour résoudre les problèmes sémantiques. De plus, le rajout de nouvelles sources modifie le schéma surtout si la source est chevauchant avec des sources intégrées. Cela demande de réécrire les relations et les requêtes au niveau du médiateur. Dans ce domaine les étapes de *mapping*, peuvent être améliorées semi-automatiquement par des méthodes et algorithmes de correspondance de schéma, aidé d'une ontologie par exemple.

```
<?xml version="1.0"?> <rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns="http://www.owl-ontologies.com/Ontology1200644993.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/Ontology1200644993.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Material">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="isParentOf"/>
            </owl:onProperty>
            <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</owl:maxCardinality>
          </owl:Restriction>
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="name"/>
            </owl:onProperty>
            <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="Mutant_plant">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Plant"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Line">
    <owl:disjointWith>
      <owl:Class rdf:about="#Plant"/>
    </owl:disjointWith>
    <rdfs:subClassOf rdf:resource="#Material"/>
  </owl:Class>
  <owl:Class rdf:ID="Mutant_line">
    <rdfs:subClassOf rdf:resource="#Line"/>
  </owl:Class>
  <owl:Class rdf:about="#Plant">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="inverse_of_isComposedBy"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="#Material"/>
    <owl:disjointWith rdf:resource="#Line"/>
  </owl:Class>
  <owl:ObjectProperty rdf:about="#isParentOf">
    <owl:inverseOf>
      <owl:FunctionalProperty rdf:ID="isChildOf"/>
    </owl:inverseOf>
    <rdfs:domain rdf:resource="#Material"/>
    <rdfs:range rdf:resource="#Material"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="isComposedBy">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:about="#inverse_of_isComposedBy"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#inverse_of_isComposedBy">
    <owl:inverseOf rdf:resource="#isComposedBy"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#name">
    <rdfs:domain rdf:resource="#Material"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">the name of a material individual</rdfs:comment>
  </owl:ObjectProperty>
  <owl:FunctionalProperty rdf:about="#isChildOf">
    <owl:inverseOf rdf:resource="#isParentOf"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
    <rdfs:domain rdf:resource="#Material"/>
    <rdfs:range rdf:resource="#Material"/>
  </owl:FunctionalProperty>
</rdf:RDF>
```

FIG. 5.17 – Description d'une partie du schéma global dans le langage OWL

```
<?xml version="1.0"?> <rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns="http://www.owl-ontologies.com/Ontology1204640380.owl#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/Ontology1204640380.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Ontology_element_class">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int" >1</owl:cardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="is_classified_into"/>
        </owl:onProperty>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="Ontology_element">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="inverse_of_is_associated"/>
        </owl:onProperty>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="Trait">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="has_a"/>
        </owl:onProperty>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="Eval_pheno">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Phenotypic_observation"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Phenotypic_observation">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="is_associated"/>
        </owl:onProperty>
        <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:ObjectProperty rdf:about="#is_associated">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:about="#inverse_of_is_associated"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="inverse_of_is_classified_into">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:about="#is_classified_into"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#has_a">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="inverse_of_has_a"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#is_classified_into">
    <owl:inverseOf rdf:resource="#inverse_of_is_classified_into"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="inverse_of_is_linked">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="is_linked"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#inverse_of_has_a">
    <owl:inverseOf rdf:resource="#has_a"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#inverse_of_is_associated">
    <owl:inverseOf rdf:resource="#is_associated"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="is_child_of">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="is_parent_of"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#is_linked">
    <owl:inverseOf rdf:resource="#inverse_of_is_linked"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#is_parent_of">
    <owl:inverseOf rdf:resource="#is_child_of"/>
  </owl:ObjectProperty>
</rdf:RDF>
```

FIG. 5.18 – Description d'une partie du schéma global dans le langage OWL (suite)

```
<?xml version="1.0"?> <rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/Ontology1204561552.owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/Ontology1204561552.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Mapping_fst">
    <owl:equivalentClass>
      <owl:Class>
        <owl:intersectionOf rdf:parseType="Collection">
          <owl:Restriction>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="is_aligned"/>
            </owl:onProperty>
            <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
          </owl:Restriction>
          <owl:Restriction>
            <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
            <owl:onProperty>
              <owl:ObjectProperty rdf:ID="inverse_of_is_mapped"/>
            </owl:onProperty>
          </owl:Restriction>
        </owl:intersectionOf>
      </owl:Class>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="Mutant_plant">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="contains"/>
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="Gene_model">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="inverse_of_have"/>
        </owl:onProperty>
        <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="Sequence">
    <owl:equivalentClass>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="inverse_of_contains"/>
        </owl:onProperty>
        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      </owl:Restriction>
    </owl:equivalentClass>
  </owl:Class>
  <owl:Class rdf:ID="Fst_in_gene"/>
  <owl:ObjectProperty rdf:about="#is_aligned">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="inverse_of_is_aligned"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#contains">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:about="#inverse_of_contains"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#inverse_of_contains">
    <owl:inverseOf rdf:resource="#contains"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#inverse_of_is_aligned">
    <owl:inverseOf rdf:resource="#is_aligned"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="is_mapped">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:about="#inverse_of_is_mapped"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#inverse_of_have">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="have"/>
    </owl:inverseOf>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#have">
    <owl:inverseOf rdf:resource="#inverse_of_have"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#inverse_of_is_mapped">
    <owl:inverseOf rdf:resource="#is_mapped"/>
  </owl:ObjectProperty>
</rdf:RDF>
```

FIG. 5.19 – Description d'une partie du schéma global dans le langage OWL (suite)

```
<view> name="Material" query="
SELECT toInteger(Null) as accession_number, name,
        toString(Null) as ecosystem, toString(Null) as remark,
        toString(Null) as type, id as id, plant_id_fk as parent_id
FROM /oryzatagline/plant
UNION
SELECT accession_number, name, remark, type, id,
        material_id_fk as parent_id
FROM /brcdb/material
UNION
SELECT toInteger(Null) as accession_number, name_line as name,
        toString(Null) as ecosystem, toString(Null) as remark,
        toString(Null) as type, id as id, toInteger(Null) as parent_id
FROM /oryzatagline/line"
</view>
<view> name="mutant_plant" query="
SELECT generation, transformation, death,
        mendelian, selected, id as id, line_id
FROM /oryzatagline/plant"
</view>
<view> name="mutant_line" query="
SELECT plant_number, id as id, construction_id
FROM /oryzatagline/line"
</view>
<view> name="line" query="
SELECT l1.description, toString(Null) as enzymatic_group, l1.id as id
FROM /oryzatagline/line l1
UNION
SELECT l2.description, l2.enzymatic_group, l2.id as id
FROM /brcdb/line l2"
</view>
<view> name="picture" query="
SELECT p1.name, p1.comment, p1.author, p1.id as id,
        p1.plant_id_fk as material_id, p1.eval_pheno_id as phenotypic_id,
        p1.construction_id
FROM (SELECT * FROM /oryzatagline/picture where author <> 'perez') p1
UNION
SELECT p2.name, p2.comment, toString(Null) as author, p2.picture_id as id,
        p2.phenotypic_id, toInteger(Null) as construction_id, p2.material_id
FROM /brcdb/picture p2"
</view>
```

FIG. 5.20 – Écriture des vues

```
<view> name="phenotypic_observation" query="
SELECT eval_date as observation_date, general_observation as
   description, plant_id as id, eval_pheno_id as phenotypic_id
FROM /oryzatagline/eval_pheno
UNION
SELECT observation_date, description
   id as id, phenotypic_id
FROM /brcdb/phenotypic_observation
</view>
<view> name="eval_pheno" query="
SELECT  developmental_stage, developmental_id,
        nb_indiv, nb_total, id as id, phenotypic_id
FROM /oryzatagline/eval_pheno"
</view>
```

FIG. 5.21 – Écriture des vues (suite)

```
TRANSCRIPT_UNIT (id, name, annotation) FST_RESSOURCES (id, name,
plant_name, source) FST_in_GENE (id, gene_id) MATERIAL
(accesion_number, name, remark , type, id) PLANT (other_name, id,
line_id) PHENOTYPIC_OBSERVATION (observation_date, description, id,
phenotypic_id)

Q2.1: select distinct(r.plant_name)
         from transcrip_unit t, fst_ressources r, fst_in_gene f
                where  t.annotation like '%ERECTA%' and t.id=f.gene_id and
                 f.id=r.id and r.source='OTL'

Q2.2: select m.name
         from material m, plant p, phenotypic_observation o
         where m.id=p.id and p.id=o.id

Q2.3 select plant_name as name from Q2.1
     intersect
     select name from Q2.2
```

FIG. 5.22 – Traitement de la requête Q2

```
MATERIAL (accesion_number, name, remark , type,id)
PLANT (other_name, id, line_id)
PHENOTYPIC_OBSERVATION (observation_date, description, id,
            phenotypic_id)
EVAL_PHENO (developmental_stage, developmental_id, nb_indiv, nb_total,
            phenotypic_id, trait_id)
TRAIT (trait, class, plant_ontology, sub_class, keywords, description, remark,
        trait_id)
ONTOLOGY_ELEMENT (id_gramene, name, definition, comment, type, synonym, sub_type,
            onto_elem_id,  phenotypic_id)

Q3: SELECT m.name
    FROM material m, plant p, phenotypic_observation
         o, eval_pheno e, trait t, ontology_element el
    WHERE m.id=p.id and p.id=o.id and t.trait_id=e.trait_id and
 ((t.class="stress" and t.sub_class="biotic") or el.name like '\%biotic stress\%')
   and e.phenotypic_id=o.phenotypic_id and el.phenotypic_id=e.phenotypic_id
```

FIG. 5.23 – Traitement de la requête Q3

Chapitre 6

Intégration de sources de données par le biais de services web

Sommaire

COMME nous l'avons constaté dans les chapitres précédents, le partage de l'information est essentiel pour la valorisation des résultats biologiques comme pour leur validation. L'intégration de sources de données hétérogènes permet d'automatiser des traitements, de découvrir de nouvelles relations entre les données et de transférer des connaissances.

Dans le chapitre 5 nous abordons l'intégration de sources à travers la médiation. Dans celui-ci nous traitons l'intégration par l'intermédiaire de services web. Pour le développement des services web, dont nous détaillerons dans la section 6.1 les principes et caractéristiques, nous avons utilisé la plateforme BioMoby. Cette dernière permet l'enregistrement et l'exécution de services web dédiés à la bioinformatique. Nous avons développé une application permettant aux biologistes d'automatiser leurs recherches à travers des sources de données distribuées et hétérogènes. Elle utilise les services web pour rechercher des informations dans les sources et rassembler les données sous une forme synthétique.

Ce chapitre comprend trois sections. Nous nous attacherons d'abord à définir les services web, les standards associés puis leurs applications dans le domaine biologique.

Dans une deuxième partie, nous décrirons le système que nous avons mis en place. Dans la section 6.2.2, nous nous appuierons sur un exemple biologique, afin d'illustrer la problématique rencontrée par les biologistes. Puis nous décrirons en section 6.2.3 les méthodes et logiciels que nous avons utilisés pour réaliser l'application.
Nous parlerons du projet BioMoby avec lequel nous avons créé les services web. Appuyés d'exemples, nous détaillerons la conception des services web ainsi que l'utilisation des outils facilitant leur développement.
Nous présenterons également, de manière détaillée, l'application que nous avons développé (section 6.2.4). Son fonctionnement sera illustré à travers un exemple qui comprend l'enregistrement d'un projet de requête et la visualisation du résultat d'exécution des services.

Enfin, dans une dernière partie, nous discuterons des améliorations pouvant être apportées à ce travail.

6.1 Les services Web

6.1.1 Définitions

Un service web (SW) ou *web service* est un programme informatique utilisé pour échanger des données entre applications hétérogènes dans un environnement distribué [W3Ca] . Ils rendent les systèmes et applications interopérables. En effet, les logiciels écris dans divers langages de programmation et sur diverses plateformes peuvent employer des services web pour échanger des données à travers les réseaux informatiques comme internet. Les SW connaissent un essor important dans l'informatique en général ils prennent de plus en plus d'importance en bioinformatique également. Ceci peut-être expliqué par l'utilisation de stan-

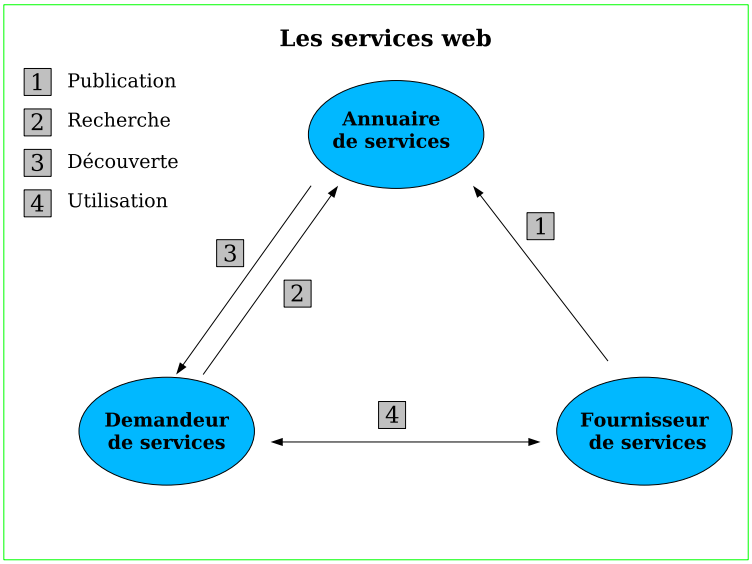


FIG. 6.1 – Schéma général des services web

dards ouverts soutenues par le W3C (World Wide Web Consortium [58]) tels que XML, HTTP et SOAP [SOA, W3Cb]. Le méta-langage XML (eXtensible Markup Language) est utilisé par les SW comme langage fondateur de représentation des données, des informations et des documents ; le protocole HTTP (High Throughput Transfer Protocol) est le protocole de transport de l'information qui garantit que les SW sont compatibles avec l'Internet public ; le standard SOAP (Simple Object Access Protocol, écrit en XML) définit un standard commun permettant à des systèmes hétérogènes de communiquer.

Dans un scénario typique de fonctionnement d'un SW (figure 6.1), un demandeur de services cherche à localiser dans un annuaire de services un SW qu'un fournisseur de services aura publié dans cet annuaire. Une fois le service identifié et localisé, le demandeur pourra invoquer le service chez le fournisseur de services. La description des services dans l'annuaire par le fournisseur, obéit au standard WSDL (Web Service Description Language) [W3Ce]. La communication avec le fournisseur de services implique l'utilisation de messages selon le protocole SOAP, ce qui sous-entend l'existence d'un serveur SOAP chez le fournisseur de services et d'un client SOAP chez le demandeur. La description d'un SW donné est essentielle car d'une part le demandeur doit pouvoir vérifier que le service répond à son besoin et d'autre part l'annuaire doit pouvoir organiser l'ensemble des SW en fonction de leurs descriptions. Le fournisseur du service doit fournir des informations sur (i) les opérations supportées, (ii) les protocoles de communication/transport sur lesquels ces opérations sont supportées, (iii) les points terminaux du réseau pour ce service (par exemple une URL d'un serveur HTTP). Un document WSDL décrit dans le standard XML les informations associées au service (concernant son interface et son implémentation) et peut être intégré dans un annuaire. Les annuaires de SW sont des structures taxonomiques organisant les services déclarés selon les besoins du domaine. Par exemple UDDI (Universal Description, Discovery and Integration) normalisé par l'OASIS[59], est un annuaire de

[58] Le World Wide Web Consortium, abrégé W3C, est un consortium fondé en octobre 1994 pour promouvoir la compatibilité des technologies du World Wide Web telles que HTML, XHTML, XML, CSS, PNG, SVG et SOAP. Le W3C n'émet pas des normes, mais des recommandations. URL :http ://www.w3.org/

[59] L'OASIS (Organization for the Advancement of Structured Information Standards) est un consortium international, distinct du W3C, qui travaille pour la normalisation et la standardisation de formats de fichiers ouverts

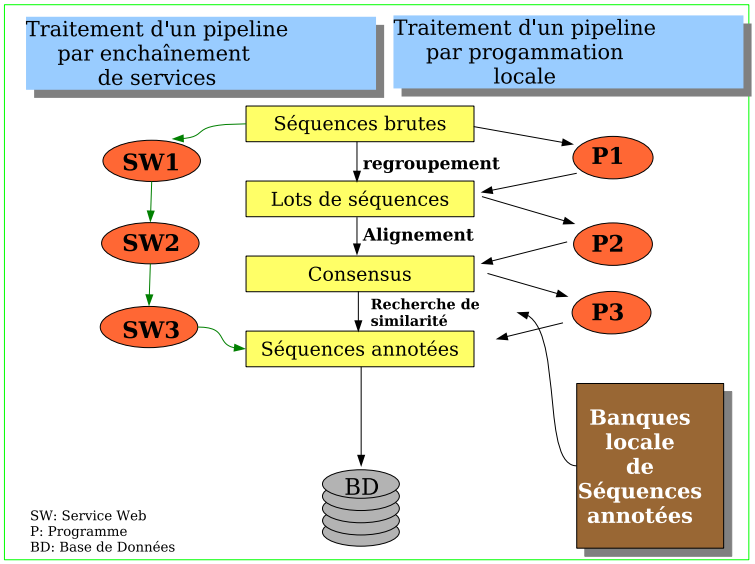


FIG. 6.2 – Enchaînement des SW

service couramment utilisé dans le domaine du commerce électronique.

6.1.2 Utilisation des Services Web dans le domaine de la biologie

Les SW se développent beaucoup dans le domaine de la bioinformatique où la diversité des sources et des outils est grande. Ils permettent (i) de répondre à un besoin d'automatisation de traitement au niveau des interfaces d'accès, là ou des programmes faisaient du "screen scraping" [60], (ii) de faire communiquer des applications hétérogènes (interopérabilité), (iii) de créer des chaînes de traitement utilisant des applications et des données distantes. Ils deviennent une solution pour les systèmes d'intégration de sources de données distribuées. Les systèmes bioinformatiques qui proposent des SW sont nombreux. Ainsi XEMBL, soaplab de l'EBI, XML central de DDBJ et Entrez utilities de NCBI ont des interfaces d'accès à leur données de séquences. Par exemple, une requête sur un Accession ID renvoie un objet biologique séquence. Le logiciel DAS (Distributed Annotation System) fourni un accès aux annotations de génomes complets lorsqu'on lui soumet un requête sur une position génomique. Pathway Database system et KEGG API fournissent un accès aux voies métaboliques à partir d'une information de séquence. Cette liste des systèmes n'est pas exhaustive, mais elle permet de constater que pour des sources utilisées fréquemment il existe déjà des SW.

L'organisation des traitements en pipeline [61] est ancrée dans la culture bioinformatique. Dès lors, l'orchestration de services est quelque chose de nécessaire pour répondre aux besoins de traitements. Par exemple, les pipelines sont très utilisés dans le domaine du traitement des séquences génomiques (figure 6.2). Lorsque des services sont développés pour chacune des étapes de l'analyse, ils doivent pouvoir être chaînés de la même manière. Toutefois, SOAP et WSDL ne suffisent pas pour permettre aux SW d'être découverts facilement par des programmes ni être

basés notamment sur XML. URL :http ://www.oasis-open.org/home/index.php

[60] On désigne par screen scraping la technique utilisée par un programme pour extraire des données à partir d'une interface d'un autre programme. http ://en.wikipedia.org/wiki/Screen_scraping

[61] Dans le jargon informatique, un pipeline est un programme informatique dont les instructions sont exécutées séquentiellement. En bioinformatique, cette méthode est utilisée enchaîner les traitements.

chaînés automatiquement avec d'autres services pour créer des pipelines [NL05]. Un certain nombre de projets proposent des solutions : BioMoby [WL02, WSEH05] , MyGrid [SRG03], Discovery Net [RKO+03] et caCORE [PCF+06, WAM+03, CHS+03].

BioMoby est un projet open source essentiellement orienté sur la découverte et l'exécution de SW biologiques. En effet par le biais d'un annuaire central, l'application propose aux fournisseurs d'enregistrer et de décrire leurs services en tenant compte d'une ontologie. L'utilisation d'une ontologie pour décrire les SW permet de faciliter la recherche et l'enchaînement des services. Toutefois BioMoby ne propose pas d'outils d'enchaînements intégrés à son API, mais des applications autonomes ont été développées pour répondre à ce besoin comme Remora [CG06] et GBrowse Moby [Wil06].

A l'inverse myGRID bénéficiant de toute l'expérience du projet Tambis (voir section 3.3.3) a développé son projet autour des enchaînements de services (*workflow*), la personnalisation et la traçabilité. L'objectif du projet est de concevoir une architecture middleware open source. myGRID possède des caractéristiques intéressantes : les résultats d'un workflow sont sauvegardés ce qui permet une consultation à chaque étape d'exécution ; la provenance des résultats est tracée (e.g. date de la dernière mise à jour d'une banque de séquence, version du logiciel BLAST, etc.) ; un service de notification est prévu pour avertir l'utilisateur que l'une des ressources utilisée a changé depuis la dernière exécution. Le point d'entrée de MyGrid est un outil de création de workflow appelé Taverna [OAF+04, HWS+06]. Sa principale caractéristique est de proposer une interface graphique à la création et l'enregistrement de workflows. Le rapprochement des projets MyGrid et BioMoby [LBW+04], permet l'utilisation de Taverna pour des services déployés sur l'annuaire central de BioMoby.

Discovery Net [RKO+03] comme MyGrid est un projet du UK e-Science Programme financé par l'Engineering and Physical Sciences Research Council (EPSRC). Bien que non spécifique à la bioinformatique, l'objectif de ce projet est le développement d'un middleware permettant de créer des enchaînements de web services réutilisables. Il a permis notamment la mise en place d'un pipeline d'annotation de génome en temps réel à partir de sources réparties.

Enfin caCORE [PCF+06] est un projet du National Cancer Institute Center of Bioinformatics (NCICB) dont le but est l'intégration de services bioinformatiques pour la recherche sur le cancer. Malgré sa spécificité de domaine, le modèle et l'architecture de caCORE ont été développés de manière générique. Le projet utilise une stratégie hybride entre un système centralisé et distribué.

6.1.3 Evolutions des standards associés aux Services Web

Malgré la forte activité qui gravite autour des SW, de nombreuses limites restent encore à franchir. La sécurité contre le piratage des systèmes est un domaine important à améliorer. En effet, les SW utilisent les mêmes protocoles que l'internet ce qui contournent les sécurités classiques mis en place à travers les *firewall*. De ce fait, beaucoup de laboratoires sont réticents à proposer des SW à cause des trous de sécurité qui peuvent être occasionnés dans leurs systèmes. Très souvent en effet, les données produites en bioinformatique sont soumises à des droits d'accès.

Les avancées sont encore faibles dans le domaine de l'orchestration de services. L'orchestration (appelé aussi la composition ou l'enchaînement), est un mécanisme capable de coordonner l'interaction des SW entre eux ainsi qu'avec leurs clients (humains ou programmes). Malgré les nombreuses propositions, il n'y a pas encore de standard émergeant dans ce domaine. Le langage BPEL4WS (Business Process Execution Language For Web Services) qui regroupe WSFL

(Web Services Flow Language) [OAS] d'IBM et XLang de Microsoft, ou encore WSCI (Web Services Choregraphy Interface) [W3Cd] sont autant de voies ouvertes par des consortiums. Récemment le W3C a proposé le langage WSCLD (Web Service Choregraphy Description Language) [W3Cc] comme candidat pour une recommandation.

Comme nous l'avons vu dans le paragraphe 6.1.2, la description sémantique des SW est nécessaire pour pouvoir effectuer une recherche automatique, une composition ou une exécution de services à travers des architectures hétérogènes. En fournissant une annotation sémantique capable d'enrichir les descriptions de services, les ontologies de services définissent la prochaine étape des SW, c'est à dire les SWS (Services Web Sémantiques). Trois approches travaillent actuellement sur les SWS : IRS-II/IRS-III [MDC03, HDM+] , OWL-S [OWL03] et WSMF [WSM].
IRS (Internet Reasoning Service) est une approche basée sur les raisonnements en ingénierie des connaissances, qui permet la réutilisation de composants.
OWL-S est une approche orientée agent, fournissant une ontologie standard pour décrire les caractéristiques des SW.
WSMF (Web Service Modeling Framework) est une approche orientée e-business, se spécialisant dans les caractéristiques e-commerce des SW, incluant sécurité et signature.

Tous ces travaux auront des conséquences positives sur la gestion de la qualité des services. En effet, il est encore difficile aujourd'hui, lorsque des services proposent les mêmes traitements d'avoir des informations indicatives (e.g. rapidité d'exécution). Par exemple, il est courant en bioinformatique d'avoir des URL différents qui proposent les mêmes ressources, comme plusieurs serveurs SRS miroirs afin de répartir les charges de travail. Pour l'instant, il n'y a aucun moyen automatique de connaître le serveur dont la banque est la plus récente parmi les 150 miroirs SRS.

6.2 Développement d'une application intégrée utilisant des services web

Afin de répondre aux besoins des biologistes, nous avons développés une application leur permettant de reproduire des recherches à travers leurs sources favorites. Cette application est basée sur les services web et permet d'automatiser les étapes successives de recherche parmi les sources biologiques. Après avoir décrit le contexte de la génomique fonctionnelle, nous illustrerons par un exemple les recherches effectuées en analyse fonctionnelle.

6.2.1 Analyse de l'existant

Dans un contexte dynamique de génomique fonctionnelle végétale, de nombreux efforts sont réalisés pour produire des ressources végétales diverses et variées. Les efforts se concentrent généralement sur l'utilisation d'espèces dites modèles. Les connaissances induites sont ensuite transférées sur d'autres espèces ayant un modèle biologique plus complexe. Impliqués dans des projets de génomique fonctionnelle, nous développons des ressources biologiques et informatiques spécifiques du riz. Parmi elles nous pouvons citer la collection de mutant d'insertion T-DNA, Tos17 et Ds. En plus des ressources génétiques que représente une telle collection, le séquençage des inserts et le phénotypage des lignées constituent les éléments essentiels des systèmes d'information que nous avons développés (e.g. OryGenesDB et OryzaTagLine, voir

section 4). Ces deux systèmes ne stockent pas uniquement nos propres données expérimentales mais intègrent également des données issues d'autres groupes et développent des outils pour aider les généticiens moléculaires dans l'obtention d'informations nécessaires à leurs analyses fonctionnelles. Dans ce domaine, nous avons créé GreenPhylDB, une ressource de génomique comparative entre les deux génomes modèles (i.e. *Arabidopsis thaliana* et *Oryza sativa*). Malgré la spécificité de leurs domaines, ces trois systèmes participent à l'enrichissement de l'information nécessaire à la découverte de la fonction des gènes. Nous allons l'illustrer par un exemple dans la section suivante.

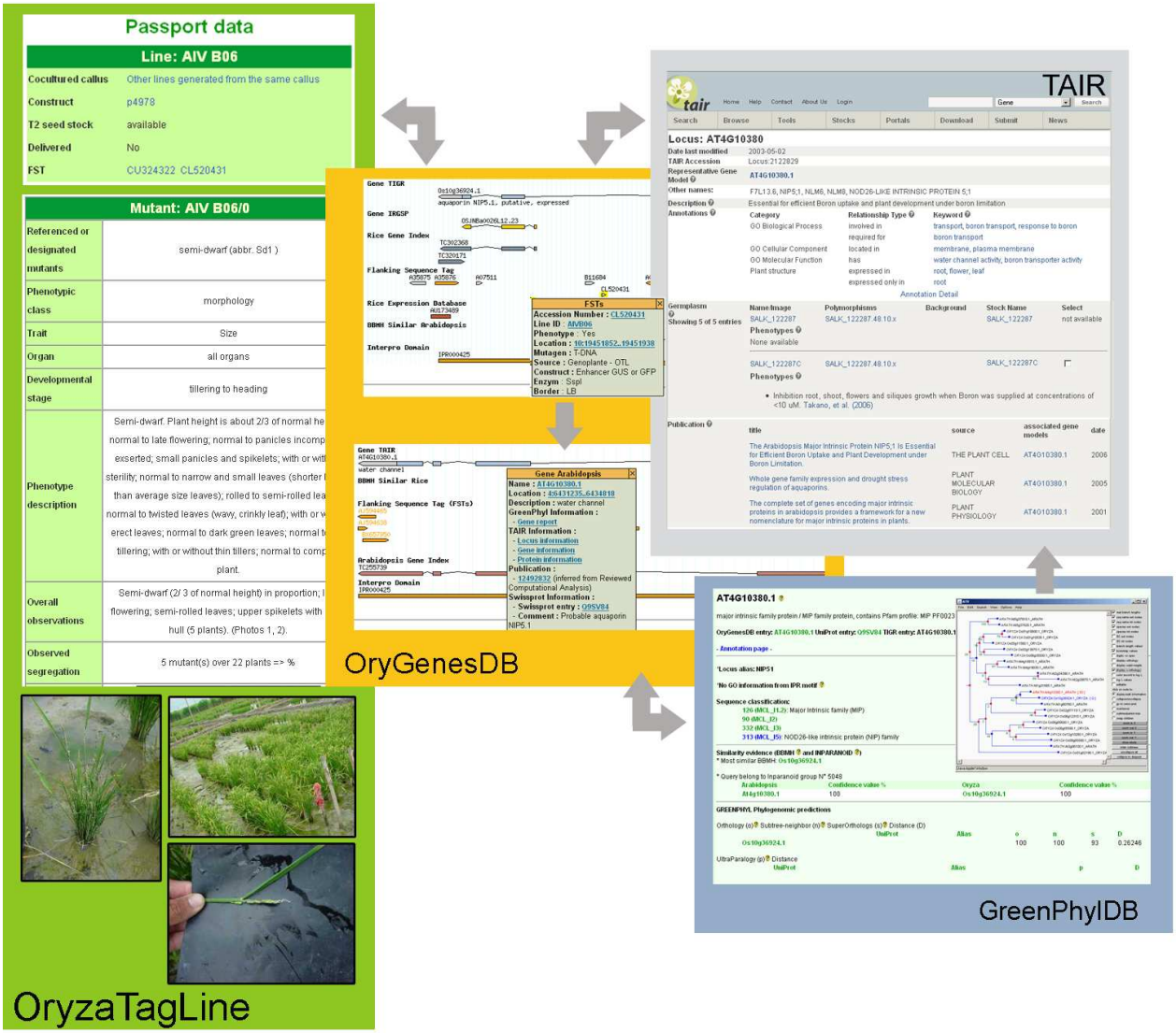


FIG. 6.3 – Navigation web entre les sources Oryza Tag Line, OrgGenesDB, GreenPhylDB et TAIR

6.2.2 Définition des cas d'utilisation

Nous allons illustrer à travers un exemple, le travail quotidien des généticiens moléculaires qui recherchent de l'information permettant d'enrichir la fonction des gènes. Cet exemple est divisé en deux parties. Dans la première nous allons montrer que la navigation parmi les sources complémentaires spécifiques du riz permet d'enrichir les données existantes sur la fonction des gènes. Dans la seconde partie, nous montrerons que l'utilisation de sources provenant d'autres espèces permet de confirmer les résultats trouvés et d'émettre de nouvelles hypothèses.

La figure 6.3 illustre la navigation entre OryGenesDB, OryzaTagLine, GreenPhylDB et TAIR. La recherche débute par le gène AT4G10380 pour le génome d'*Arabidopsis* sur l'interface d'OryGenesDB (génome *Arabidopsis*). Cette interface apporte une première indication sur sa fonction de transport (i.e. water channel). A partir d'un lien hypertexte, il est possible de voir dans la source GreenPhylDB les relations d'orthologie qu'a ce gène avec le génome du riz. En effet, des gènes orthologues peuvent être trouvés avec cette base qui utilise des méthodes phylogénétiques pour identifier les relations entre les deux génomes. Le gène orthologue Os10g36924.1 est identifié. Grâce aux liens http, il est possible de voir dans la base OryGenesDB sa fonction biologique (i.e. aquaporin), sa position sur le génome du riz ainsi que toutes les couches d'annotations qui sont liées à cette position. La couche "Flanking sequence tag" (FST) permet de vérifier si des mutants d'insertion ont été identifiés. Dans ce cas, la séquence de cette FST, CL520431, indique que la lignée AIVB06 de la collection Génoplante a été observée. Pour avoir plus de détails sur les observations phénotypiques disponibles, les biologistes utilisent un lien hypertexte basé sur le nom de la plante. Ce lien les dirige vers la source OryzaTagLine. Parmi toutes les observations réalisées sur cette lignée, la source affiche un phénotype semi-nain avec ses images associées.

Par cet exemple nous pouvons constater que la navigation dans des sources de données complémentaires permet d'avoir de l'information qui enrichie la fonction du gène. Ici, chaque source spécifique du riz a permis de préciser la fonction du gène Os10g36924.1.

Dans la deuxième partie de cet exemple nous allons montrer que la redondance d'informations similaires permet de confirmer des résultats. En effet, des informations supplémentaires peuvent être apportées par des ressources de différentes espèces, par exemple, la ressource TAIR, qui centralise les données d'*arabidopsis thaliana*.

Dans la figure 6.3, elle est accessible grâce à un lien http à partir de la source GreenPhylDB. Sur la fiche du gène AT4G10380.1, la description de la fonction du gène permet de comprendre que la fonction de transport intra cellulaire est liée au métabolisme du bore. De plus, le gène a un rôle dans le développement cellulaire, ce qui peut expliquer le phénotype de semi-nanisme observé chez les mutants de riz. En bas de page sont indiquées des publications à partir desquelles ces informations sont basées. A la lecture des expériences décrites dans ces publications, les biologistes obtiennent des éléments détaillés sur la fonction du gène ainsi qu'une base de travail leur permettant de confirmer cette hypothèse.

Les informations ainsi recueillies sur la ressource TAIR permet d'apporter de nouvelles données et d'émettre de nouvelles hypothèses de travail. Pour confirmer l'hypothèse selon laquelle, le phénotype développemental observé est bien lié au métabolisme du bore, le biologiste devra adapter les expérimentations réalisées sur Arabidopsis pour les effectuer sur le riz.

6.2.3 Matériels et méthodes

Nous avons pu constater qu'il est important de relier l'information à travers plusieurs sources de données. Afin d'articuler cette information nous avons choisi de créer des services Web en utilisant la plateforme BioMoby.

6.2.3.1 Description de la plateforme BioMoby

Architecture de BioMoby Le projet MOBY est partagé en deux sous-projets : MOBY-S pour *moby services* et S-MOBY pour *semantic moby*. Poursuivant les objectifs d'intégration et d'interopérabilité, le premier projet se focalise sur la centralisation de services bioinformatiques dans un annuaire central alors que le deuxième projet s'oriente sur une architecture de médiation basée sur le modèle pivot RDF. Afin de développer des services nécessaires aux analyses biologiques, nous avons choisi le projet BioMoby services pour la dynamique qu'il bénéficie dans le domaine végétal. Ce projet open source a pour objectif d'être au service de la découverte, l'intégration et l'interopérabilité des bases de données biologiques. La figure 6.4 illustre les différentes étapes d'enregistrement (1), de découverte (2), et d'exécution (3) d'un service Web. En premier lieu, le service Web doit être enregistré sur l'annuaire MOBY Central qui contient des informations sur les types de données, de services et les relations entre les types de données (1). Un client, peut alors retrouver un service sur cet annuaire et en demander l'adresse fournisseur (2). Avec cette adresse, le client ira directement se connecter chez le fournisseur (c'est-à-dire sur le serveur comportant le code du service Web) pour appeler le service en lui fournissant un fichier XML comprenant des données d'entrées (3). Le client attendra en retour les données de sorties comportant les résultats sous la forme d'un autre fichier XML (3). D'une façon similaire, le client peut appeler et exécuter plusieurs autres services Web, en fournissant en entrée d'un second service, le fichier XML de sortie du premier service, et créer ainsi un enchaînement.

Les ontologies de BioMoby La "connaissance" du central s'exprime sous forme d'ontologies. En premier lieu, une ontologie hiérarchise les services par le type d'action qu'ils exécutent (ontologie de service). Une seconde structure les noms de domaine (namespace) avec lesquels sont enregistrés les services. Enfin, une ontologie structure les types de données (data types) qui sont utilisés en entrée et en sortie d'un service (ontologie de classe).

- Le type de service fait référence à son action. Dans la plupart des projets, les services sont de type Retrieval, c'est-à-dire qu'ils ont vocation à récupérer des informations dans des sources de données. Cette ontologie comprend aussi Analysis (pour un service pratiquant une analyse), ou encore Conversion (pour une conversion de format). L'ontologie de service est formalisée par des graphes de type "RDF-like". Pour les ontologies de services, les noeuds sont les types des analyses ou des transformations (par exemple au niveau supérieur de l'ontologie : "analysis", "parsing", "retrieval", etc.) et les relations sont de type "isa".
- Le nom du domaine de définition des données (namespace) permet d'éviter des conflits entre services utilisant la même ontologie.
- Les data types sont les objets utilisés en entrée et en sortie des services Web. Ce sont concrètement des classes Java (dans notre cas mais ils peuvent être écris dans d'autre langages) qui caractérisent les données envoyées et retournées par les services. L'ontologie de classe est formalisées par des graphes de type "RDF-like". Pour celle-ci, les noeuds

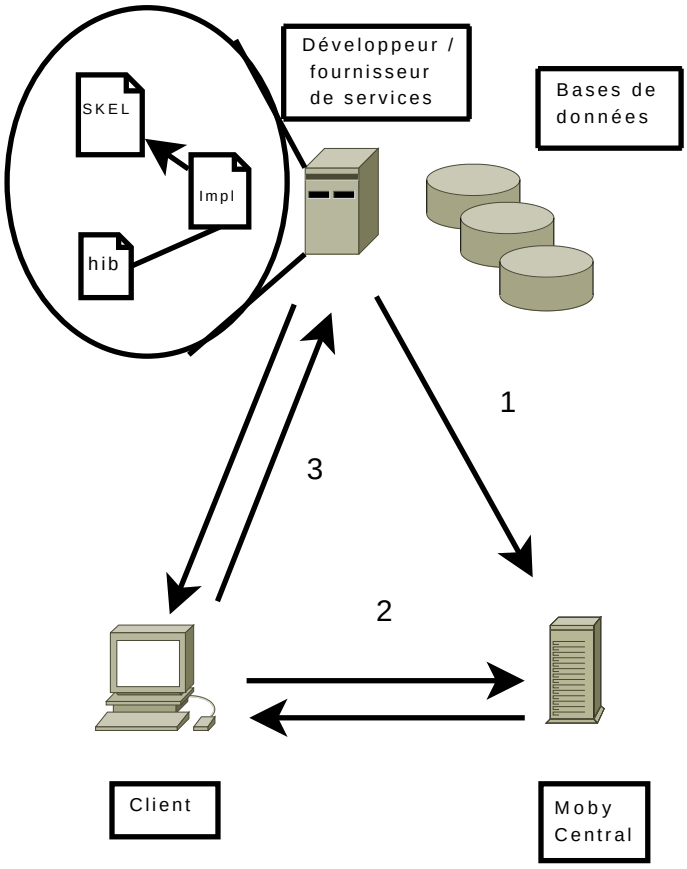


FIG. 6.4 – Schéma de fonctionnement d'appel de services Web via BioMoby.

sont les noms de classes, les relations sont de type "isa", "hasa" et "has". Les structures des classes sont définies par des schémas XML.

La figure 6.5 illustre la description d'un service Web, telle que l'on peut la trouver par exécution d'un simple client. Ces informations, additionnées du nom de service et de son adresse URL, sont retrouvées lors de l'appel de découverte du service (étape 2 sur la figure 6.4) et constituent les informations nécessaires et suffisantes à l'exécution d'un service Web. La figure 6.6 correspond aux données d'entrée fournies au service décris dans la figure 6.5. Parmi les éléments qui composent ce fichier, nous pouvons reconnaître ceux qui définissent le nom du service, le data type le namespace et enfin la valeur envoyée.

```
Name:           getGeneIdByLocation
Type:            Retrieval
Category:      moby
Auth:             orygenesdb.cirad.fr
Desc: Take a location (ex : 1:1..20000) and returns a list of gene identifiers.
The service works with the 2 species (Oryza sativa or Arabidopsis
thaliana) according to the namespace given (OryGenesDB_Oryza or
OryGenesDB_Arabidopsis).
URL:http://sat.cirad.fr/sat/Orygenesdb/services/getGeneIdByLocation
Contact:        orygenesdb@cirad.fr

Primary inputs:
   Name:    location
    Data Type:
             Name:    GCP_SimpleIdentifier
         Namespaces:
                  Name:    OryGenesDB_Oryza

                  Name:    OryGenesDB_Arabidopsis
Outputs:
   Name:    gene_id
   Elements in collection:
         Data Type:
                  Name:    GCP_SimpleIdentifier
              Namespaces:
                       Name:    OryGenesDB_Oryza

                       Name:    OryGenesDB_Arabidopsis
```

FIG. 6.5 – Description d'un service Web de type BioMoby.

6.2.3.2 Conception des services web

L'enregistrement des services sur le central registry de BioMoby peut se faire de plusieurs manières. En effet, BioMoby dispose des API Perl et Java pour toutes les étapes d'enregistrement des services, namespaces et objets. L'API Java (jMoby) dispose notamment d'un client

```
<moby:MOBY xmlns:moby="http://www.biomoby.org/moby">
  <moby:mobyContent>
    <moby:mobyData moby:queryID="sip_1_">
      <moby:Simple moby:articleName="location">
        <moby:GCP_SimpleIdentifier moby:id="" moby:namespace="OryGenesDB_Oryza">
          <moby:String moby:id="" moby:namespace=""
            moby:articleName="name">1:1..20000</moby:String>
        </moby:GCP_SimpleIdentifier>
      </moby:Simple>
    </moby:mobyData>
  </moby:mobyContent>
</moby:MOBY>
```

FIG. 6.6 – Description d'un service Web de type BioMoby.

graphique nommé Dashboard, qui permet toutes ces étapes d'enregistrement de service ainsi que des outils de visualisation afin de naviguer dans les différentes ontologies ou de tester les services.

L'étape suivante dans la création de services, consiste à les implémenter. L'interface MoSeS Generator du logiciel BioMoby Dashboard (figure 6.7) permet de générer les classes Java correspondantes aux data types, ainsi que plusieurs classes nécessaires aux traitements et créations d'objets Moby. Dashboard permet également de générer une classe comprenant le " squelette " du service Web à implémenter (cette classe est nommée : nom_du_serviceSkel.java). Elle contient principalement des méthodes pour créer un objet, récupérer les données d'entrées, et créer un fichier XML de sortie (voir plus bas). Par convention, les fonctionnalités ajoutées aux services ne se font pas au niveau de cette classe mais dans une classe qui implémente celle-ci (e.g. nom_du_serviceImpl.java). C'est au niveau de la méthode *processIt* que sont implémentées les principales fonctionnalités, comme par exemple les requêtes effectuées sur des bases de données.

- *createFromXML()* créé un objet mobyInput à partir du fichier XML d'entrée des données. Paramètres : (data, "GCP_SimpleIdentifier").
- *prepareOutput()* créé un objet mobyOutput. Paramètres : (mobyInput).
- *processIt()* est la méthode la plus importante du programme. Elle exécute les requêtes, les traitements et vérifications sur les données d'entrées et sorties. Paramètres : (mobyInput, mobyOutput).
- *toXML()* permet de créer un fichier XML de sortie à partir de l'objet mobyOutput.
- *get_location()* récupère les données d'entrées d'un objet GCP. Paramètres : (MobyJob request).
- *set_gene_idSet()* retourne l'objet GCP choisi contenant les résultats. Paramètres : (MobyJob response, GCP_SimpleIdentifier[] values).

6.2.3.3 L'enchaînement des services web

En général, les services web pour qu'ils puissent être réutilisables, exécutent des actions atomiques (e.g. renvoyer une séquence nucléique à partir d'un numéro d'accession). Les applications bioinformatiques nécessitent d'enchaîner ces actions pour créer une dynamique dans

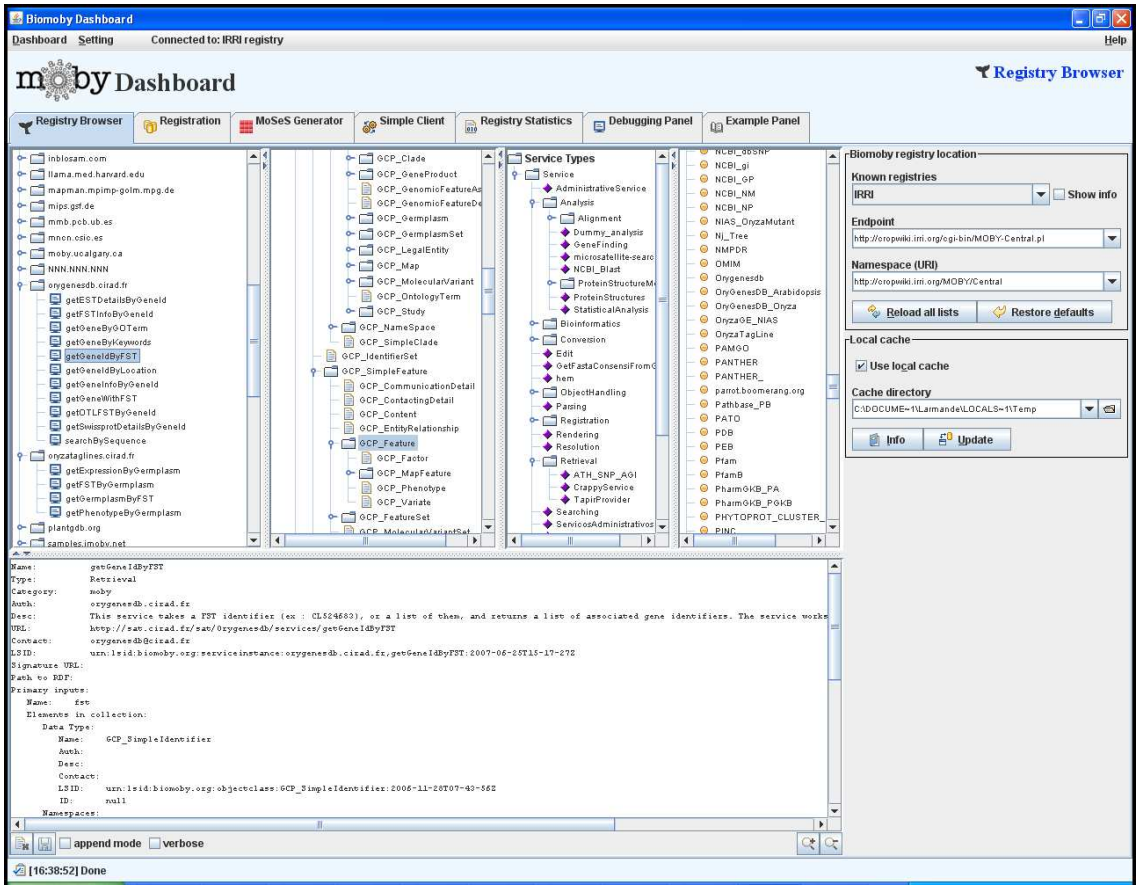


FIG. 6.7 – Présentation de BioMoby Dashboard

les traitements de données. Les workflows de services ont pour fonction d'enchaîner ces programmes les uns après les autres tout en gérant la transmission de données.

Des applications ont été développées autour de BioMoby pour pouvoir enchaîner les services web. Il y a tout d'abords GBrowse MOBY, développé par les concepteurs de la plateforme BioMoby [Wil06]. L'application web permet d'exécuter et d'enchaîner des services de type MOBY mais pas de manière automatique. Le projet Rémora [CG06] est également un application web qui permet la composition de workflow à partir de services MOBY. Le projet a la spécificité de pouvoir sauvegarder les workflows et les partager parmi les utilisateurs. Enfin, Biowep [RBB+07] est une application web qui permet d'exécuter des workflows prédéfinis composés à partir des services MOBY et MyGrid. L'application permet également de soumettre des workflows développés avec un logiciel comme Taverna.

Utilisation de Taverna L'enchaînement de services web peut être réalisé avec un logiciel comme Taverna. Cette application permet aux biologistes ou bioinformaticien de construire, sans grande connaissances en programmation, des analyses complexes sur des ressources publiques ou privées. Initialement conçu dans le cadre du projet MyGrid (voir section 6.1.2) le rapprochement des projets MyGrid et BioMoby [LBW+04], permet d'utiliser Taverna pour des services déployés sur l'annuaire central de BioMoby.

6.2.4 Résultats

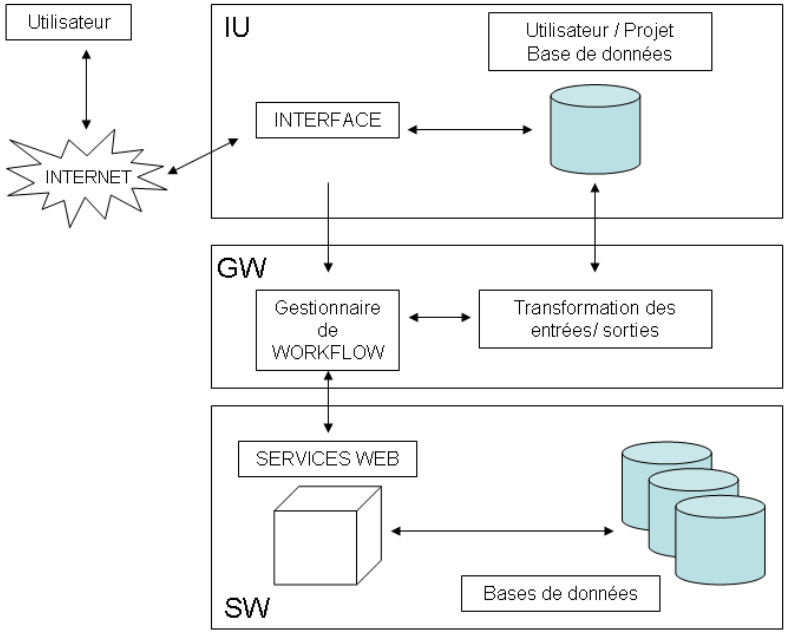
Nous avons développé une application intégrée permettant aux utilisateurs d'enregistrer et d'exécuter des recherches sur des sources distribuées par le biais de services web. L'architecture de cette application est décrite dans la figure 6.8. Elle comprend trois blocs. Le bloc "service web" est la couche adaptateur qui permet à l'application d'interroger les sources de données. Le bloc "gestionnaire de workflow" permet d'effectuer l'enchaînement des services web ainsi que la transformation des résultats. Le bloc "interface utilisateur" concerne les différentes fonctions qui permettent aux utilisateurs de s'inscrire, gérer leurs projets et voir les résultats. Dans les sections suivantes nous allons détailler le développement de l'ensemble des blocs.

6.2.4.1 Création des services web

Dans l'application que nous avons développée, les services web vont reproduire les différentes requêtes effectuées par les sources pour extraire de l'information. Pour chacune d'entre elles nous avons enregistré un service sur le central registry en utilisant BioMoby Dashboard. Afin de réaliser la connexion avec les sources de données et de créer le service nous devons définir les data types correspondant aux objets biologiques utilisés.

Définition des data types Dans la mesure où les data types Moby sont structurés dans une ontologie, nous avons réutilisé des data types existants. Dans notre cas, les data types définis dans le cadre d'un projet international, le Generation Challenge Programme[62] (GCP) ont été réutilisés pour nos services. Ceci nous permet d'avoir des services web sémantiquement compatibles avec ceux développées dans le cadre du projet GCP.

[62]GCP est un programme de recherche international qui vise à mieux utiliser la diversité présente dans les banques de gènes, au service de l'amélioration variétale : http ://www.generationcp.org/



IU: Interface utilisateur
GW: Gestionnaire de workflow
SW: Services Web

FIG. 6.8 – Architecture de l'application intégrée

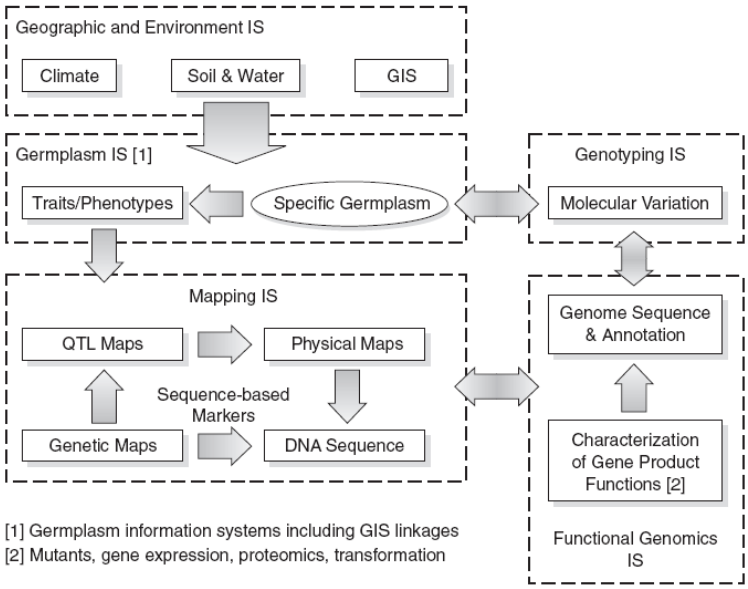


FIG. 6.9 – Illustration du Modèle CGP (extrait de Bruskiewich et al, [BDH+06])

Description du Génération Challenge Programme Le GCP est un consortium international de recherche qui oeuvre pour l'amélioration des plantes utiles aux pays en développement. Le consortium facilite le transfert des avancées technologiques dans les multiples domaines de la biologie vers les pays dépourvus de ressources. Parmi les grands objectifs, l'accent est mis sur l'amélioration des problèmes de résistance des plantes à la sécheresse et aux maladies. Dans ce contexte, le consortium a mis en place une plateforme informatique dont l'objectif, à long terme, est de permettre l'échange d'information et de ressources [BDH+06]. Pour cela, le GCP a modélisé son domaine tout en réutilisant des modèles existant comme Chado pour la génomique [MEC07] et FUGE pour la génomique fonctionnelle [JPS+06]. Comme le montre la figure 6.9, de nombreux domaines sont représentés (i.e. génotypage, génomique, cartographie génétiques, géographie). Compte tenu du consensus de l'approche, la réutilisation de tels objets à travers des data types BioMoby déjà réalisés est intéressante pour notre application.

Le choix des Data types a représenté une étude approfondie du modèle GCP[63]. En effet, le projet GCP contient de multiples classes Java, traduites du modèle de représentation, pour exprimer les données biologiques. Une correspondance des objets du modèle avec les data types Moby a été récemment mise à disposition [64]. Nous avons identifié le paquetage illustré en figure 6.11 comme modèle générique pouvant contenir nos types de données. Trois Data types différents ont été choisis en fonctions de nos besoins :

- l'objet unique GCP_SimpleIdentifier est utilisé pour caractériser les données locus, nom de lignées, localisation chromosomique et domaine protéique (représentation d'un objet GCP_SimpleIdentifier en figure 6.6),
- L'objet GCP_Feature est utilisé lorsqu'un service est destiné à renvoyer une collection de plusieurs informations à partir d'une entrée ; par exemple, pour un locus donné, on souhaite renvoyer sa position sur le chromosome, son sens sur le brin, sa fonction...
- L'objet GCP_Value est un objet hérité de GCP_Feature. Un GCP_Feature peut donc

[63] Site web du GCP : http ://pantheon.generationcp.org/
[64] Mapping entre le modèle GCP et le data type BioMoby : http ://pantheon.generationcp.org/moby/

```
<moby:GCP_Feature moby:id="Os01g10900.1"
moby:namespace="OryGenesDB_Oryza">
  <moby:GCP_Value moby:id="pseudochromosome_id" moby:articleName="values">
    <moby:String moby:articleName="name">1</moby:String>
  </moby:GCP_Value>
  <moby:GCP_Value moby:id="start_model" moby:articleName="values">
    <moby:String:articleName="name">5811583</moby:String>
  </moby:GCP_Value>
  <moby:GCP_Value moby:id="end_model" moby:articleName="values">
     <moby:String:articleName="name">5816009</moby:String>
  </moby:GCP_Value>
  <moby:GCP_Value moby:id="strand_tu" moby:articleName="values">
     <moby:String moby:articleName="name">-</moby:String>
  </moby:GCP_Value>
  <moby:GCP_Value moby:id="com_name" moby:articleName="values">
    <moby:String moby:articleName="name">ATP binding protein, putative,
       expressed</moby:String>
  </moby:GCP_Value>
  <moby:GCP_Value moby:id="feat_name_tu" moby:articleName="values">
    <moby:String:articleName="name">12001.t00963</moby:String>
  </moby:GCP_Value>
  <moby:GCP_Value moby:id="feat_name_model" moby:articleName="values">
    <moby:String moby:articleName="name">12001.m07710</moby:String>
  </moby:GCP_Value>
</moby:GCP_Feature>
```

FIG. 6.10 – Description d'un service Web de type BioMoby.

contenir plusieurs GCP_Value, ce qui est utile dans notre cas, car chaque GCP_Value représentera une information (position chromosomique, sens du brin, fonction...), toutes rassemblées sous l'autorité d'un seul GCP_Feature (représentatif d'un locus par exemple ; voir figure 6.10).

Connexion aux bases de données Pour les services qui effectuent une connexion à des bases de données (e.g. OryGenesDB, Oryza Tag Line, GreenPhylDB) plusieurs moyens sont envisagés. Par exemple, une connexion peut être réalisée avec l'API JDBC qui permettra d'encapsuler directement des requêtes SQL dans le service web. Mais il y a également des applications qui permettent de gérer la persistance des objets en bases de données relationnelles. Hibernate[65] est une application open source qui a ses caractéristiques. Elle permet de créer des objets qui sont connectés à la structure et aux données d'une base de données relationnelle. Hibernate existe en tant que plugin Eclipse[66]. La création d'objets spécifiques d'une source nécessite la configuration de fichiers de configuration (figure 6.12) et de mapping (figure 6.13). Dans le premier

[65] Hibernate : http ://www.hibernate.org/

[66] Eclipse est un environnement de développement intégré open source principalement dédié au développement en Java

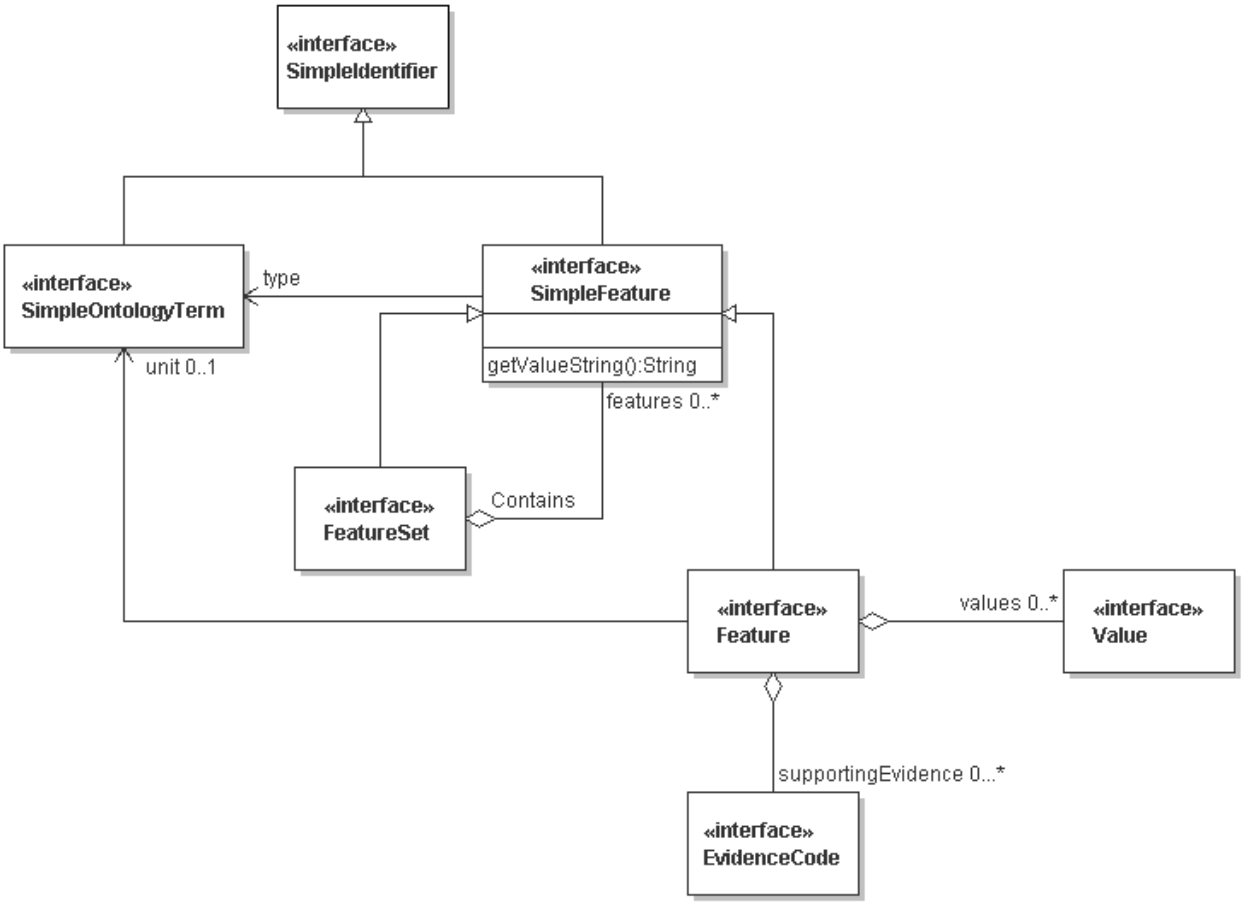


FIG. 6.11 – Partie du modèle GCP utilisée dans notre application

sont répertoriés les paramètres de connexion, les drivers spécifiques du RDBMS[67] ainsi que les références vers les fichiers de mapping. La figure 6.13 représente les correspondances entre les colonnes de la base de données et les attributs des classes. Chaque classe contient un élément identifiant <id> et plusieurs éléments <property>. En plus d'effectuer le mapping ces derniers permettent de typer et contraindre les attributs. Non représenté dans ce cas, les relations entre classes peuvent être représentées dans le fichier de mapping. La figure 6.14 présente un exemple de requête pouvant être effectuée sur une base de données en utilisant Hibernate. Dans ce cas la requête est effectuée avec la méthode find qui renvoi tous les enregistrements. Par la suite, le code permet de naviguer dans les enregistrements pour n'afficher que le numéro du chromosome grâce à la méthode *getChromosome()*. Un exemple de service web développé est donné en annexe A. Dans la figure 6.4 nous pouvons voir comment les différents fichiers sont organisés sur le serveur du fournisseur afin d'exécuter un service web.

```
<hibernate-configuration>
    <session-factory>
        <!-- local connection properties -->
        <property name="hibernate.connection.url">
            jdbc:mysql://valois.cirad.fr/ORYGENESDB_PUBLIC
        </property>
        <property name="hibernate.connection.driver_class">
            org.gjt.mm.mysql.Driver
        </property>
        <property name="hibernate.connection.username"></property>
        <property name="hibernate.connection.password"></property>
        <!-- dialect for MySQL -->
        <property name="dialect">
            net.sf.hibernate.dialect.MySQLDialect
        </property>
        <property name="hibernate.transaction.factory_class">
            net.sf.hibernate.transaction.JDBCTransactionFactory
        </property>
        <mapping resource="Pseudochromosome.hbm" />
        <mapping resource="Dna.hbm" />
    </session-factory>
</hibernate-configuration>
```

FIG. 6.12 – Exemple de fichier hibernate de configuration

Synthèse des services web développés Une liste de 14 services Web a été établie pour répondre aux besoins du projet. Le tableau 6.1 détaille, pour chaque service créé et implémenté, la base de données visée (source), les données que prennent en entrée les services (input), ainsi que leur sorties (output). Le terme *gene report* signifie qu'une collection de plusieurs informations différentes va être renvoyée. Il correspond aussi aux 8 services Web terminaux, c'est-à-dire ceux qui renvoient des résultats dont les données seront ensuite intégrés dans un seul et même fichier XML.

[67] RDBMS : relational database management system

N°	Nom du service Web	Source	Entrée (input)	Sortie (output)
1	getGeneInfoByGeneId	OryGenesDB	Gene id(s)	Gene report
2	getFSTInfoByGeneId	OryGenesDB	Gene id(s)	Gene report
3	getSwissprotDetailsByGeneId	OryGenesDB	Gene id(s)	Gene report
4	getESTDetailsByGeneId	OryGenesDB	Gene id(s)	Gene report
5	getGeneIdByLocation	OryGenesDB	Location	Gene id(s)
6	getOTLFSTByGeneId	OryGenesDB	Gene id(s)	FST id(s)
7	getGeneIdByGO	OryGenesDB	Go id	ene id(s)
8	getGeneIdByFST	OryGenesDB	FST id	Gene id(s)
9	getPhyloGenomicInformationByGeneId	GreenPhyl	Gene id(s)	Gene report
10	getInterproDetailsByGeneId	GreenPhyl	Gene id(s)	Gene report
11	getGeneIdByInterpro	GreenPhyl	IPR id	Gene id(s)
12	getExpressionByGermplasm	OryzaTagLine	Germplasm id	Gene report
13	getPhenotypeByGermplasm	OryzaTagLine	Germplasm id	Gene report
14	getGermplasmByFST	OryzaTagLine	Germplasm id	FST id(s)
15	getFSTByGermplasm	OryzaTagLine	FST id	Germplasm id
16	getGenesByKo	KEGG	KO id	Gene id(s)
17	getGenesByEnzyme	KEGG	Enzyme id	Gene id(s)

TAB. 6.1 – Liste des services web développés

Développement de clients d'appel des services web Les services web sont des programmes indépendants qui fonctionnent grâce à un serveur web spécifique. Dans notre cas les services ont été installés sur un serveur apache tomcat[68] avec axis[69] pour pouvoir exécuter les services. Ces programmes peuvent être invoqués de différentes manières, par exemple avec Dashboard, ou bien enchaînés avec Taverna, Rémora ou GBrowse Moby. Dans notre cas, nous avons inclus ces clients dans une interface web existante qui était réalisée en perl.

La figure 6.15 représente un exemple de client invoquant le service getGeneIdByLocation. Le programme fait appel à une API perl spécifique à BioMoby ainsi que des parseurs XML. Les variables MOBY_SERVER et MOBY_URI indiquent l'endroit où se situe le central registry. Dans ce cas, les services ne sont pas directement enregistrés sur le central registry BioMoby d'origine, mais sur un central du domaine GCP. Ce client effectue plusieurs actions : il recherche un service dont le nom est donné en paramètre(1), en récupère ses paramètres (2), et fait appel au service (4) avec une donnée d'entrée (3). Le client utilise des packages développés dans le cadre du projet BioMoby (5).

6.2.4.2 Développement de workflows

Les services web que nous avons développés, sont conçus de manière atomique afin qu'ils puissent être réutilisables. La figure 6.16 représente l'organisation de ces services entre eux avec les différentes données qu'ils traitent en entrée et sortie. Selon les cas d'utilisation que nous avons définis au préalable, quatre entrées biologiques différentes permettent d'obtenir un maximum d'information se rapportant aux gènes annotés. Ces entrées, en effet, sont les points

[68] Apache Tomcat est un conteneur de servlet J2EE issu du projet Jakarta

[69] Axis est un package Java issu du projet Apache dont le but est de fournir une API de développement et d'execution de services web

d'entrée de 4 workflows potentiels. Tous les résultats des services terminaux (1-4,8,9,11,12) participent au résultat final.

Utilisation de Taverna La figure 6.17 représente la conception des workflows de services réalisés avec Taverna. L'analyse de l'exécution des workflows conçus avec Taverna montre qu'ils ne sont pas tous fonctionnels. En effet, le *parser* de data type CGP_Feature fourni pour Taverna n'est pas fonctionnel. Une alternative à ce problème aurait été de modifier nos services web pour utiliser un autre data type. Mais compte tenu d'autres remarques des utilisateurs, nous nous sommes orientés vers une solution adaptée. En effet, pour certains biologistes, l'utilisation de Taverna reste encore complexe et les concepts qu'il manipule sont abstraits. De plus l'exécution des workflows fonctionnels est assez lente. Enfin, la visualisation des résultats issus des workflows est brute, de ce fait elle complique la synthèse de l'analyse biologique. Pour ces raisons, nous nous sommes orientés vers la conception d'un gestionnaire de workflows adapté aux besoins du projet.

Conception d'un gestionnaire de workflow Nous avons réalisé le gestionnaire dans un langage pouvant être interprété par un serveur web de sorte qu'il puisse être intégré dans une interface web dédiée (voir section 6.2.4). Les principales caractéristiques devant être implémentées dans le gestionnaire sont l'amélioration du temps d'exécution, la gestion des exécutions (e.g. multi-utilisateurs, exécution sur mise à jour des sources), traitement et organisation des résultats, mise en cache des résultats. Le gestionnaire doit permettre également aux utilisateurs d'accéder facilement à l'exécution de leurs workflows, ainsi qu'à la visualisation des résultats. Une des principales problématiques à ce niveau, se situe dans la présentation des données. En effet, celles-ci sont trop importantes pour les présenter de façon directe à l'utilisateur de plus dans le format de SOAP/BioMoby. Un traitement des résultats doit donc être effectué permettant à l'utilisateur d'avoir accès aux résultats de manière synthétique dans un premier temps puis de manière détaillée. En résumé, les cas d'utilisation retenus pour la conception du gestionnaire sont la gestion d'exécution, le traitement des résultats ainsi que leur accès.

La figure 6.18 illustre les actions effectuées par le gestionnaire. La partie A concerne l'exécution des services alors que la partie B concerne le traitement des résultats. Pour le premier cas de fonctionnement de l'outil (A), il s'agit uniquement d'un appel à un workflow, et donc aux services Web qui le composent. Pour cela, la variable contenant le type de données d'entrées (locus, région chromosomique, domaine interpro ou identifiant de plante) va déterminer quel workflow exécuter. Les données d'entrées vont être insérées dans la construction d'un objet BioMoby (par la fonction déjà présente dans le client générique). Cet objet est encapsulé dans un fichier XML et directement envoyé aux services Web qui le consomment. Le programme attend le retour des services Web sous forme de fichiers XML contenant les résultats. Les informations de ces fichiers XML sont alors stockées. Enfin, le programme parcours tous les fichiers pour créer le fichier XML unique par projet (ou workflow).

L'exécution des workflows, et a fortiori les appels de services Web sont des procédures nécessitant un temps d'action important. La figure 6.19 montre la séquence de fonctionnement d'un workflow. Dans ce cas, on distingue bien les différents services exécutés. L'étape limitante dans cette partie est la répétition des appels vers le central. Cet inconvénient a nécessité de chercher un système permettant de paralléliser ces appels, au lieu d'exécuter les services de façon linéaire l'un après l'autre. En effet, le schéma des workflows (visible en figure 6.16) permet de voir que certains services n'ont pas besoin d'attendre que le précédent soit terminé,

et peuvent donc tout à fait s'exécuter en simultané. C'est dans cette optique que nous avons utilisé le module Perl Parallel : :ForkManager, qui permet d'exécuter plusieurs processus en parallèle. Cet outil, particulièrement adapté pour une optimisation du temps d'exécution de notre programme, a permis de réduire par deux le temps d'exécution.

Afin de créer un fichier unique, une DTD a été définie (voir annexe B). L'objectif de cette dernière étant de valider le document XML final généré par le programme. Lorsque tous les services ont été exécutés le programme parcours les résultats et les compile tous dans un fichier XML unique. La racine du document est un élément <projet>. Toutes les informations obtenues se rattachent à un locus.

Dans une deuxième fonction, il s'agit de présenter les résultats sous une forme synthétique, de façon à rendre lisible les données d'un projet, car la quantité d'information est trop importante pour tout afficher. Pour cela, le programme va se servir du fichier XML unique créé. Ce fichier contient des balises spécifiques indiquant la présence ou l'absence d'information pour les identifiants de gènes retenus dans le projet. A ce niveau, le programme génère alors un tableau contenant les paramètres du projet (identifiant de gène, localisation chromosomique et fonction), ainsi que la présence ou l'absence d'autres informations (pour chaque gène du projet).

Pour la dernière fonction, le gestionnaire génère une fiche complète (e.g. tableau) pour chaque identifiant de gène du projet. Cette fiche rassemble toutes les informations pertinentes que les services Web ont pu extraire des trois systèmes d'informations.

6.2.4.3 Implémentation de l'interface Web utilisateur

Afin de rendre l'utilisation de nos services Web facilement accessibles aux utilisateurs, une interface graphique a été développée. Cette interface s'est ajoutée au site déjà existant d'OryGenesDB. La création de l'interface Web permet aux utilisateurs de créer un espace personnel de navigation en enregistrant leurs informations personnelles et leurs projets. La définition des projets permet de renseigner les informations nécessaires à l'exécution des workflows (e.g. le nom d'espèce, les types de données et les données d'entrées). La présentation des résultats doit aussi tenir compte du grand nombre d'informations disponibles, et de leur affichage syntaxique à l'écran. L'interface doit permettre d'accéder à la liste des projets, les consulter, les supprimer et les mettre à jour.

Pour développer cet outil, une base de données a été implémentée sous MySQL. Elle est utilisée pour stocker les multiples informations des utilisateurs et les paramètres de chacun des projets. Elle est utilisée également lors de la navigation sur l'interface (e.g. vérification du compte, démarrage d'une session personnelle, utilisation de quelques informations personnelles pour certaines fonctions comme l'envoi de message email). La base est également utilisée au lancement d'un service Web, lorsque le gestionnaire de workflow interroge cette base pour retrouver les informations d'entrées nécessaire à l'appel des services.

Nous avons choisi d'illustrer les résultats par un exemple de projet de requête. L'étape d'enregistrement (figure 6.20) de projets intervient après s'être identifié ou enregistré. L'utilisateur a la possibilité de fournir des données sous forme de liste selon le type d'entrée sélectionnée. Dans ce cas nous sélectionnons une zone sur le chromosome 1 du riz. Avec l'interface de gestion des projets (figure 6.21), il est possible de vérifier l'état d'exécution d'un projet, d'effectuer une mise à jour ou une suppression. Le projet "Region1" qui vient d'être créé n'est pas encore lancé, alors que les deux projets suivant ont été exécutés. Les résultats se visualisent d'abord sous la forme d'un tableau de synthèse (figure 6.22) avec des liens vers les informations dé-

taillées (figure 6.23). Dans la vision synthétique, les résultats sont structurés par gène avec une partie concernant leur localisation et leur fonction. S'ajoute à cela une partie indiquant la présence ou l'absence d'information enrichissant la fonction du gène (e.g. EST, phenotype, etc). La vision détaillée permet d'avoir un aperçu graphique de la région chromosomique ainsi que des informations détaillées tel que la liste des FST, le nom des plantes ayant un phénotype mutant observé, etc. Lorsque des références croisées sont disponibles, des liens http font pointer les informations vers la source d'origine.

6.3 Discussion

Cet outil d'intégration répond bien aux besoins des biologistes. En effet, il facilite le travail des biologistes en ne proposant qu'une seule interface de navigation. De plus, le fait d'enregistrer des projets permet de garder une trace des recherches effectuées. Enfin, les interfaces de visualisation de résultats permettent de gagner du temps dans les analyses et mettent en valeurs des données pertinentes. Par ailleurs, de nombreuses fonctionnalités facilitent la mise à jour des résultats, leurs sauvegardes ou leurs exports. Il y a toutefois des améliorations qui peuvent être apportées.

Sur le plan biologique, des services peuvent être ajoutés afin d'enrichir les informations concernant les gènes. Par exemple l'ajout des services développés pour l'espèce arabidopsis dans le cadre du projet Planet peut enrichir l'information d'orthologie fournie par la source GreenPhylDB. Des informations sur les références bibliographiques peuvent être ajoutées également si des services sont disponibles pour interroger la source Pubmed et que des liens de références croisées existent au sein des sources. Tous ces services peuvent être inclus dans l'enchaînement des services existants si les correspondances sont disponibles (e.g. nom de gène - pubmed id)

Nous avons relevé, la nécessité de mettre en place, un moyen d'alerter les utilisateurs lorsqu'une mise à jour est effectuée sur les sources impliquées dans leurs workflows. L'implémentation de cette fonctionnalité est contraignante : en effet quels critères doit on définir pour indiquer au service la mise à jour ? Que spécifie BioMoby dans ce domaine ? Pour l'instant rien n'est envisageable du coté de l'annuaire central BioMOBY. Alors, la solution retenue a été de mettre en place une action de mise à jour réalisée par l'utilisateur. Ne sachant pas comment vont évoluer les langages et les APIs que nous utilisons, nous projetons de mettre en place un système automatique de lancement de workflow, configurable également par l'utilisateur. Le résultat du workflow (document XML final) pouvant être comparé au résultat précédent, ce qui déclencherait une alerte.

Les améliorations pouvant être apportées concernent également l'application. Pour l'instant les utilisateurs ne sélectionnent pas les workflows à executer. Un effort peut être porté sur cette interactivité. Par exemple, une liste de workflows, incluant également des workflows disponibles sur d'autres serveurs (e.g. myGRID, Remora) peut être proposée en fonction des données d'entrée soumises. Les utilisateurs aurait alors le choix de sélectionner un ou plusieurs workflows à exécuter. Les résultats de ces workflows devant être intégrés dans les fichiers XML finaux, cela nécessiterait d'étendre la DTD actuellement conçue pour les workflow proposés. De la même manière et compte tenu de la spécificité des data types utilisés dans l'application (e.g. ontologie CGP), il est possible de proposer , une liste de services GCP compatibles accessibles sur le MoBY central registry compatibles avec les données soumises en entrée.

```
<hibernate-mapping package="hibernate.orygenesdb">
    <class name="Pseudochromosome" table="PSEUDOCHROMOSOME">
        <id
            column="pseudochromosome_id"
            name="Id"
            type="integer"
        >
            <generator class="increment" />
        </id>
        <property
            column="chromosome_length"
            length="10"
            name="ChromosomeLength"
            not-null="false"
            type="integer"
         />
        <property
            column="assembly"
            length="10"
            name="Assembly"
            not-null="false"
            type="integer"
         />
        <property
            column="centromere"
            length="10"
            name="Centromere"
            not-null="false"
            type="integer"
         />
        <property
            column="chromosome"
            length="10"
            name="Chromosome"
            not-null="false"
            type="integer"
         />
    </class>
</hibernate-mapping>
```

FIG. 6.13 – Exemple de fichier hibernate de mapping pour la classe Pseudochromosome

```
import java.util.*;
import net.sf.hibernate.*;
import com.orygenesdb.hibernate.*;

public class ChomosomeList {

 public static void main(String[] args)
    throws HibernateException {

     Session session = HibernateUtil.currentSession();

     List list = session.find("from Pseudochromosome");
     Iterator it = list.iterator();
     while(it.hasNext())
     {
          Pseudochromosome chromosome = (Pseudochromosome)it.next();
        System.out.println(chromosome.getChromosome());
     }

     HibernateUtil.closeSession();
 }
}
```

FIG. 6.14 – Exemple de code java utilisant une connexion hibernate pour la classe Pseudochromosome

```
use lib '/usr/local/lib/perl5/site_perl/5.8.7';
use XML::XPath;
use XML::XPath::XMLParser;
use MOBY::Client::Central;
use MOBY::Client::Service;
# (5) # package BioMoby pour interagir avec le service
$ENV{MOBY_SERVER} = "http://cropwiki.irri.org/cgi-bin/MOBY-Central.pl";
$ENV{MOBY_URI} = "http://cropwiki.irri.org/MOBY/Central";

my $Central = MOBY::Client::Central->new();
my $choice = shift or die "Choose :\n - getGeneIdByLocation\n";
my ($authURI,$serviceName,$namespace,@param,$articleName ,$param) ;
# (3) # exemple d'un jeu de paramètres pour le service getGeneIdByLocation
if ($choice eq "getGeneIdByLocation") {
     $authURI = 'orygenesdb.cirad.fr';
     $serviceName = 'getGeneIdByLocation';
     $namespace = "OryGenesDB_Oryza";
     @param = ("1:1..10000");
     $param = "1:1..20000";
     $articleName = "location";
}

# (1) # recherche du service
my ($service, $r) = $Central->findService(authURI => $authURI, serviceName =>
$serviceName);
$service = shift @{$service};
# (2) # récuperation de la definition WSDL du service
my $wsdl = $Central->retrieveService($service);
my $serviceInstance = MOBY::Client::Service->new(service => $wsdl)
%or die "Web Sservice Not found\n";
# (3) # appel à la création de l'objet d'entrée
my $query = template($namespace,$articleName,$param);
# (4) # exécution du service
my $result = $serviceInstance->execute(XMLinputlist =>
[[$articleName, $query]]);
print $result ."\n";
# (5 ) # fonction de création de l'objet BioMoby d'entrée sub
template {
  my ($namespace,$articleName,$param) = @_;
  my $query = "
  <moby:GCP_SimpleIdentifier moby:id=\"\" moby:namespace=\"$namespace\">
  <moby:String moby:id=\"\" moby:namespace=\"\"
  moby:articleName=\"name\">$param</moby:String>
  </moby:GCP_SimpleIdentifier>";
  return $query;
}
```

FIG. 6.15 – Exemple de client d'appel de service web en perl

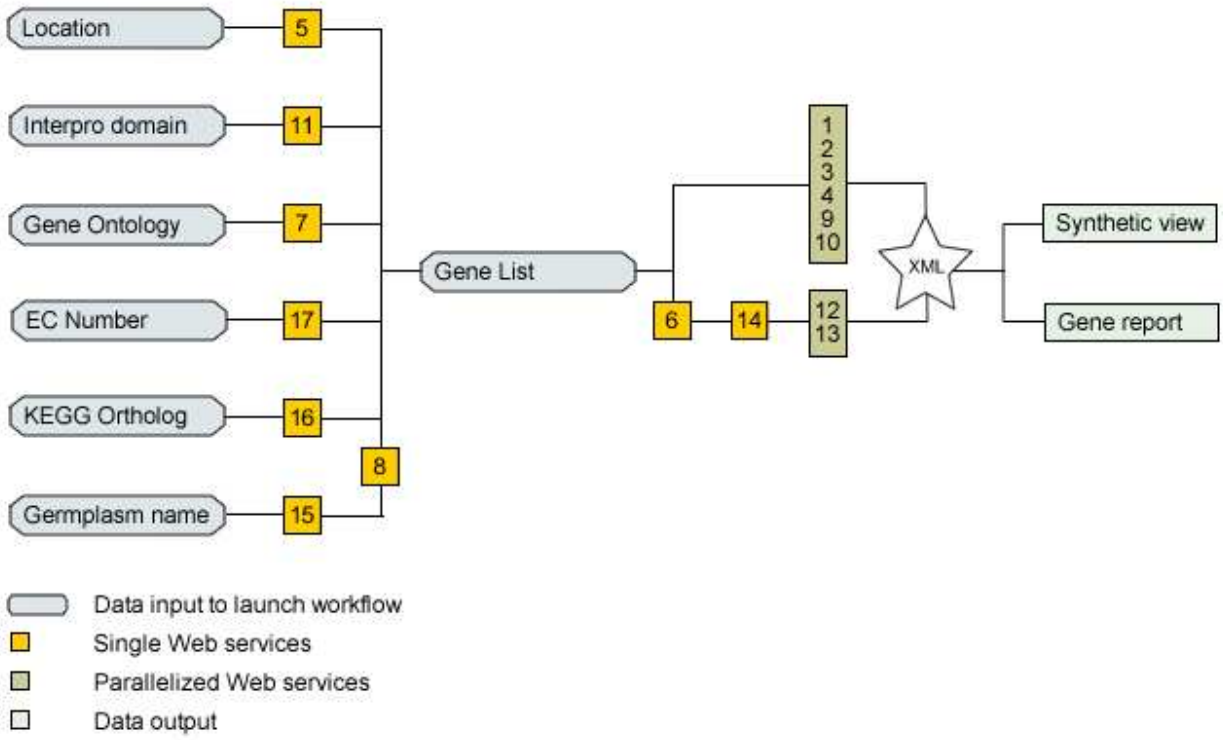


FIG. 6.16 – Schéma des workflows de services créés

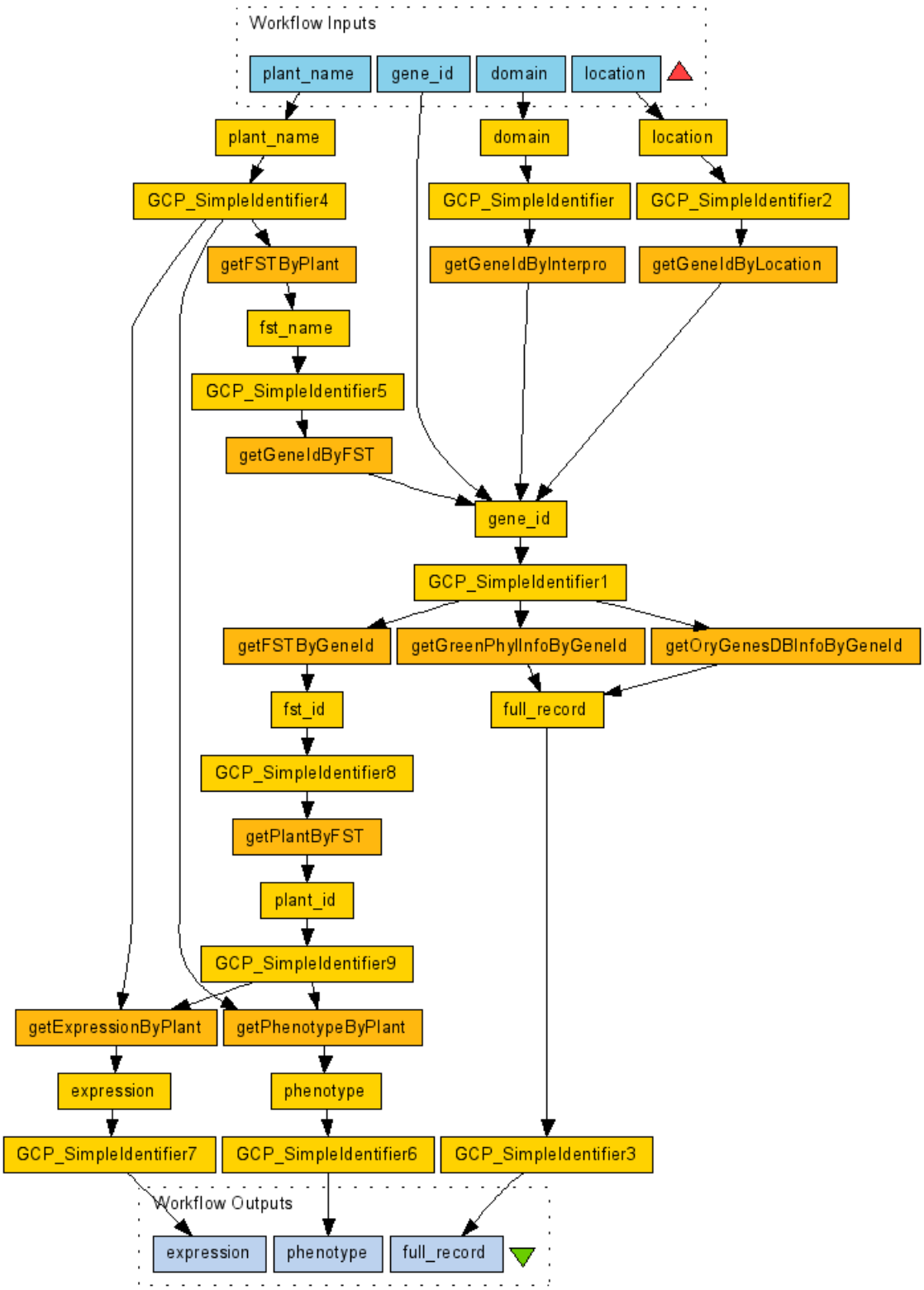


FIG. 6.17 – Schéma des workflows de services créés avec Taverna

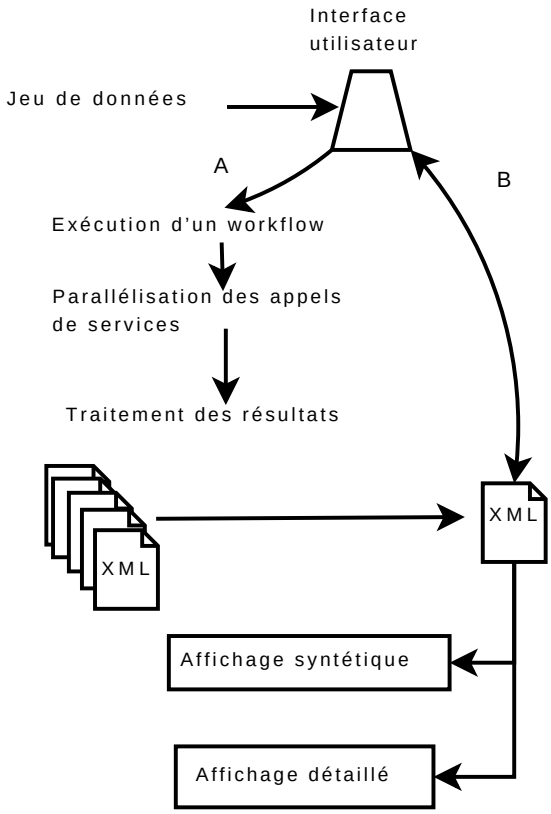


FIG. 6.18 – Schéma des actions effectuées par le gestionnaire de workflows

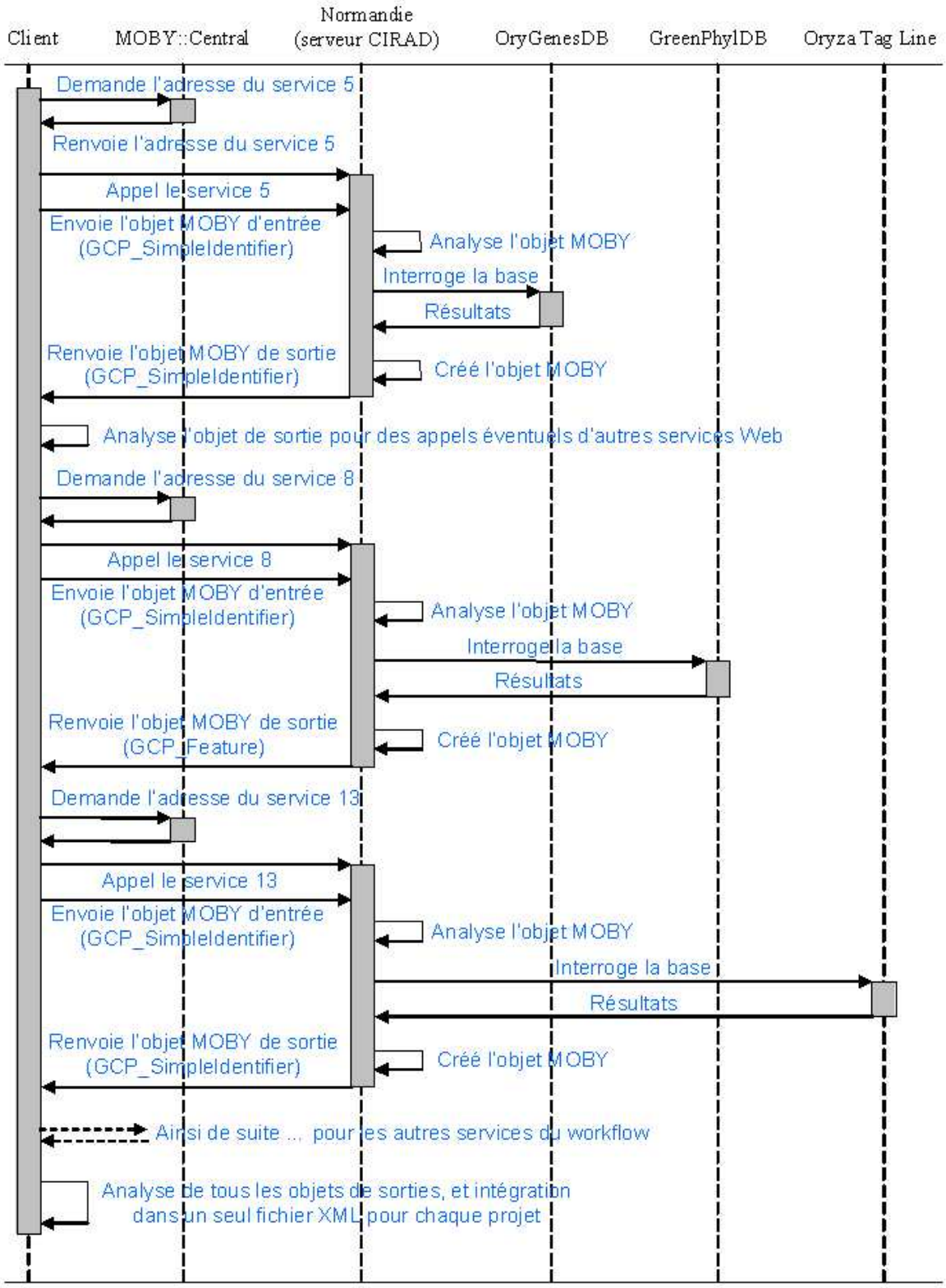


FIG. 6.19 – Diagramme de séquence représentant l'exécution d'un workflow

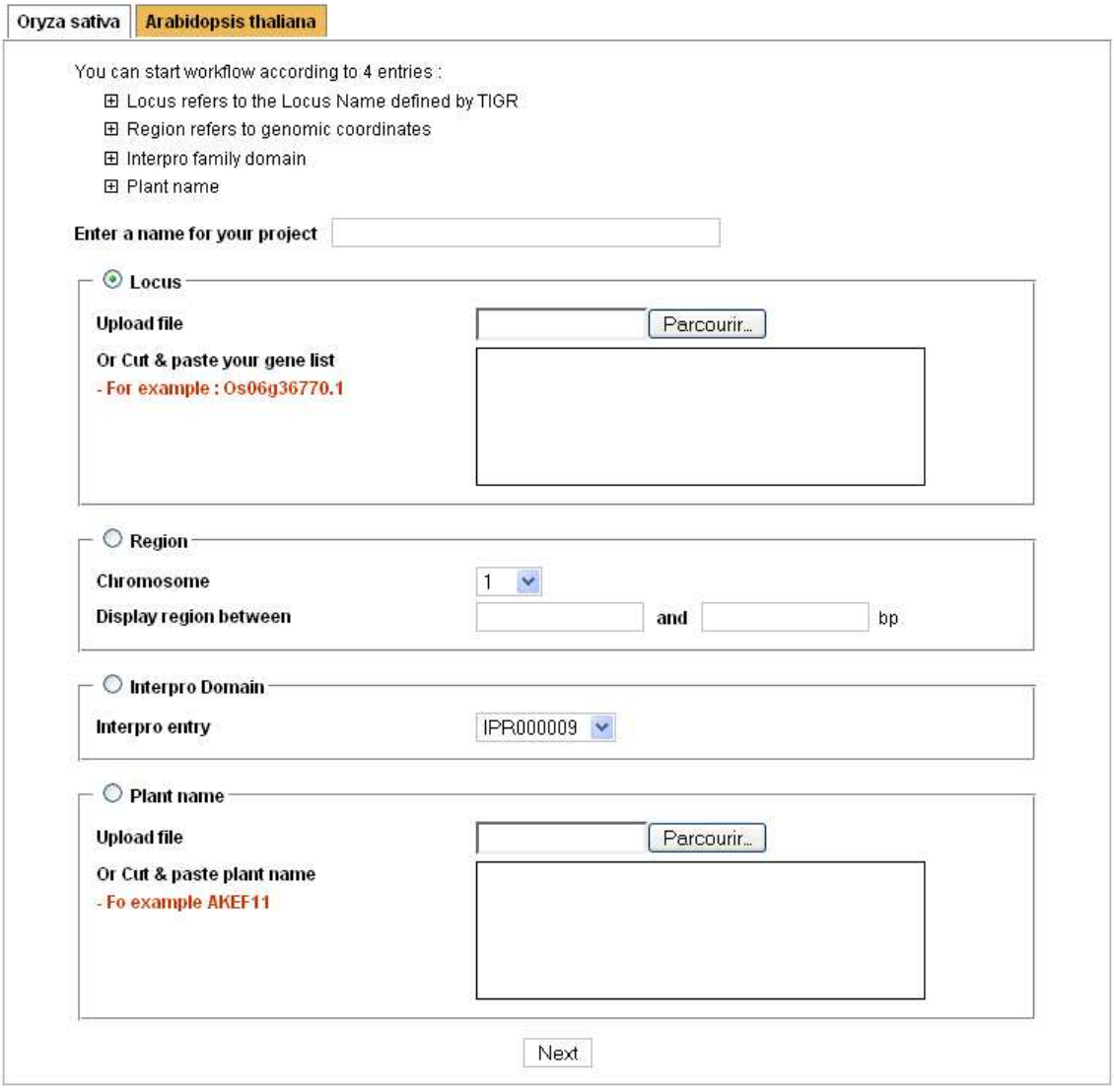


FIG. 6.20 – Interface d'enregistrement d'un projet de requête

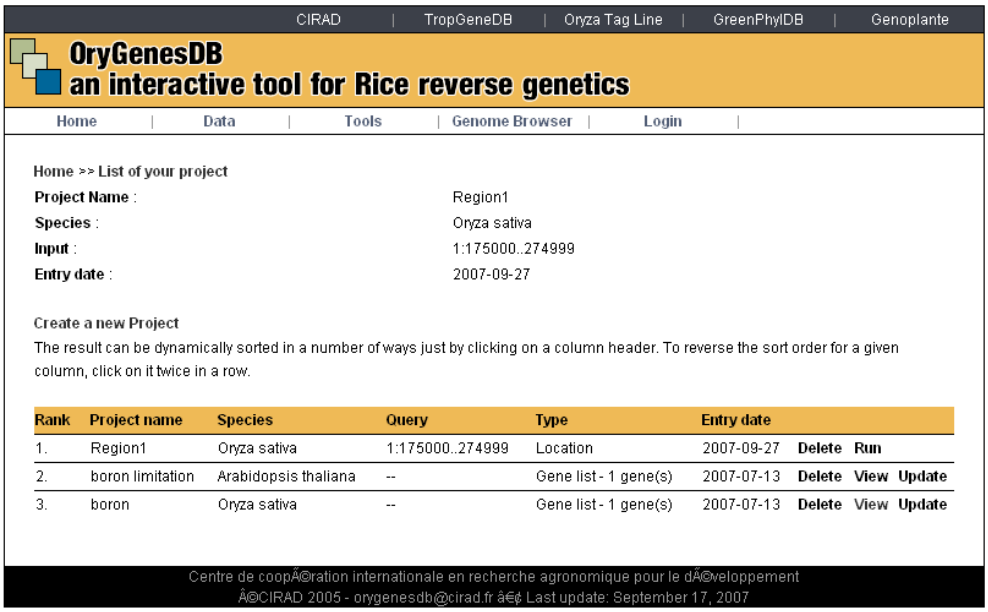


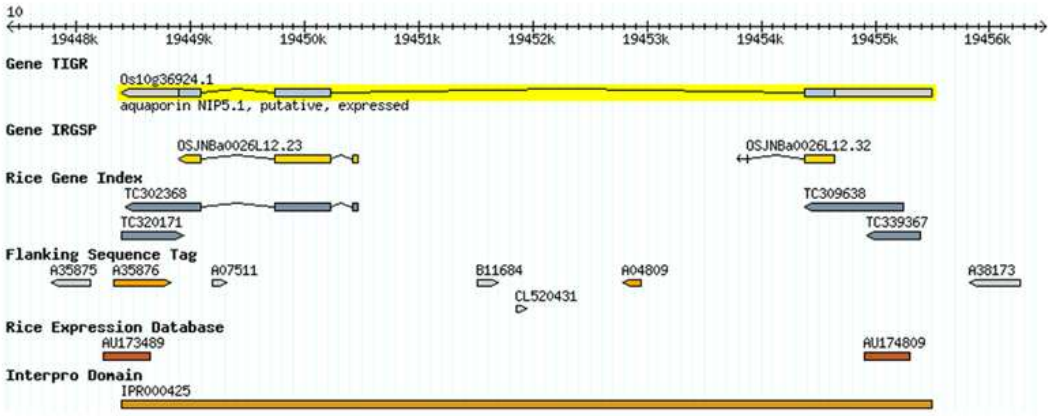
FIG. 6.21 – Interface de gestion des projets

Gene Id	Chr.	Start	End	Function	Have FST	Have Expression observed	Have Phenotype observed	Is Supported by EST/cDNA
Os01g01360.1	1	175112	177614	peptide transporter PTR2, putative, expressed	Yes	No	No	Yes
Os01g01369.1	1	183048	187648	3-beta-hydroxysteroid-delta-isomerase, putative, expressed	Yes	No	No	Yes
Os01g01380.1	1	190236	193089	expressed protein	Yes	No	No	Yes
Os01g01390.1	1	194836	197605	expressed protein	No	No	No	Yes
Os01g01390.2	1	194836	197605	expressed protein	No	No	No	Yes
Os01g01390.3	1	194836	197605	expressed protein	No	No	No	Yes
Os01g01390.4	1	194717	197605	expressed protein	No	No	No	Yes
Os01g01390.5	1	194717	197605	expressed protein	No	No	No	Yes
Os01g01400.1	1	198746	203545	expressed protein	Yes	Yes	No	Yes
Os01g01410.1	1	202512	206408	leucine-rich repeat receptor protein kinase EXS precursor, putative, expressed	Yes	Yes	No	Yes
Os01g01410.2	1	202512	206408	leucine-rich repeat receptor protein kinase EXS precursor, putative, expressed	Yes	Yes	No	Yes
Os01g01420.1	1	207046	210988	expressed protein	Yes	Yes	No	Yes
Os01g01420.2	1	207046	210988	expressed protein	Yes	Yes	No	Yes
Os01g01430.1	1	212928	214459	NAC domain-containing protein 18, putative, expressed	Yes	No	No	Yes
Os01g01440.1	1	215813	217578	hypothetical protein	Yes	No	No	Yes
Os01g01450.1	1	223699	226131	stress responsive protein, putative, expressed	Yes	No	No	Yes
Os01g01450.2	1	223699	226131	stress responsive protein, putative, expressed	Yes	No	No	Yes
Os01g01460.1	1	229179	229490	retrotransposon protein, putative, Ty3-gypsy subclass	No	No	No	Yes
Os01g01470.1	1	238668	240270	NAC domain-containing protein 77, putative, expressed	Yes	No	No	Yes
Os01g01484.1	1	245509	253680	light-mediated development protein DET1, putative, expressed	Yes	Yes	No	Yes
Os01g01484.2	1	245509	253680	light-mediated development protein DET1, putative, expressed	Yes	Yes	No	Yes
Os01g01484.3	1	246019	253680	light-mediated development protein DET1, putative, expressed	Yes	Yes	No	Yes
Os01g01484.4	1	245509	253086	light-mediated development protein DET1, putative, expressed	Yes	Yes	No	Yes
Os01g01484.5	1	247419	253680	light-mediated development protein DET1, putative, expressed	Yes	No	No	Yes
Os01g01500.1	1	256409	256711	conserved hypothetical protein	No	No	No	Yes
Os01g01510.1	1	258306	265238	expressed protein	Yes	No	No	Yes
Os01g01510.2	1	258332	265273	expressed protein	Yes	No	No	Yes
Os01g01520.1	1	266948	271886	anthranilate N-benzoyltransferase protein 1, putative, expressed	Yes	Yes	No	Yes
Os01g01530.1	1	273087	274683	transposon protein, putative, CACTA, En/Spm sub-class	Yes	No	No	Yes

FIG. 6.22 – Interface de résultats synthétiques

1. Gene Report

Locus Name	Os10g36924.1
Chromosome	10
Start	19448392
End	19455495
Function	aquaporin NIP5.1, putative, expressed
Gramene information	Gene report Transcript report Exon report Protein report



2. FST Report

FST name	Germplasm	Source	Type Insert	Location	Orientation
CL520431	AIVB06	OTL	T-DNA	Intron	reverse

3. Phenotype Report

Germplasm	Mutant name	Abreviation	Phenotypic class	Trait Name	Trait ontology	Organ	Description	Keywords
AIVB06	semi-dwarf	Sd1	Morphology	Size	TO:0000207	all organs	Semi-dwarf. Plant height is about 2/3 of normal height.	
AIVB06	aberrant panicle organization-1	apo1	Morphology	Architecture	TO:0000089 TO:0000436	panicle	Panicle architecture and floral organs abnormal; high sterility.	
AIVB06	Abnormal spikelets	abns	Panicle traits	Abnormality	TO:0000657 TO:0000436	spikelet	Malformed spikelets; seed sterility	
AIVB06	Bad tipped panicle	mup	Panicle traits	Abnormality	TO:0000089	panicle	Malformed upper part of panicles; with abnormal spikelets	

4. Phylogenomic predictions (Orthology)

View result on GreenPhyl

Gene id	Orthology	Subtree-neighbor	Super Orthologs	Distance
At1g80760.1	7	100	0	0.39535
At4g10380.1	100	100	93	0.26246

5. cDNA FL and/or EST Report

EST / cDNA name	Description	Source
TC302368	homologue to UP|Q9ATN1_MAIZE (Q9ATN1) NOD26-like membrane integral protein ZmNIP3-1, partial (75%)	OGI
AU173489	AU173489 Rice root Oryza sativa (japonica cultivar-group) cDNA clone R3335, mRNA sequence	RED

6. Interpro Report

IPR name	Type	Description	Start	End	Score
IPR012269	Family	Aquaporin	87	290	1.9e-58
IPR000425	Family	Major intrinsic protein	87	281	6e-100

FIG. 6.23 – Interface de résultats détaillés

Troisième partie

Synthèse et discussion

Chapitre 7

Synthèse et discussion

Sommaire

DANS une première partie, nous faisons une synthèse récapitulative de nos travaux. Cette synthèse est motivée par l'apparente disparité de chacune des activités menées, les unes portant sur la conception et la réalisation de bases de données à visée intégrative en génomique végétale, les autres sur l'élaboration de deux démarches d'intégration indépendantes. Un état des lieux s'impose, d'abord pour rappeler chaque grande étape réalisée au sein de ces activités et notre degré de satisfaction face à chacune de ces étapes , ensuite pour démontrer les liens de forte dépendance entre toutes ses activités et enfin pour commencer à en dégager les éléments conceptuels et/ou techniques qui nous semblent les plus remarquables. Dans une deuxième partie, nous présentons successivement les éléments de conclusion portant sur nos deux expériences acquises dans le domaine de l'intégration de données biologiques et qui se sont révélées riches en enseignement. Notre principale contribution à ce sujet porte sur des lignes de conduite claires que les communautés de biologistes se doivent de suivre lorsqu'il s'agit d'exploiter au mieux données et outils de traitement acquis par tous. Enfin, nous terminons par une projection sur l'avenir et notamment sur les efforts à produire afin de proposer aux biologistes du CIRAD un système d'intégration de sources de données végétales parfaitement opérationnel.

7.1 Synthèse

Le tableau ci-dessous dresse un récapitulatif des systèmes développés dans le courant de la thèse. Différents aspects, qu'ils soient de nature conceptuelle, méthodologique ou encore technique, sont considérés. Chaque système a fait l'objet de choix conceptuels/ méthodologiques/ techniques spécifiques et répond également à des objectifs spécifiques. Il est à noter que la notion d'intégration est déjà présente, de manière plus ou moins latente, dans les bases de données OryGenesDB et OryzaTagLine, et est ensuite au centre des préoccupations des systèmes définis à partir de Le Select et des services Web. Différentes indications concernant la prise en charge de l'intégration sont fournies. Des notions qui relèvent de la performance des systèmes sont également reportées. Il s'agit notamment des politiques mises en place pour la mise à jour des données, des mécanismes d'optimisation des requêtes, de l'expressivité des modèles de données ou encore des fonctionnalités facilitant l'interaction avec l'usager au niveau de chacune des interfaces

Système d'intégration	Tâches de Conception	Réalisations et Modèle Architectural	Politique de Mise à Jour	Producteur et/ou Fournisseur de Données	Type d'intégration	Mécanismes d'Optimisation de Requêtes	Flexibilité de l'Interface	Expressivité du langage de requête	Interface utilisateur
OryGenesDB	Rétro-conception à partir du MPD GMOD (modèle partagé GBROWSE) et extension du modèle en fonction des besoins spécifiques du projet	Architecture 3 tiers ; Couches métiers : perl ; Couche de persistance : BDR ; Interface : perl/HTML/Javascript	mise à jour annuelle, inféodée au changement de la source de données : TIGR Gene Indices	Producteur de données et fournisseur données provenant d'autres sources	Matérialisée ; Spécifique d'un domaine	Création d'index uniques et non uniques sur attributs porteurs de contraintes de clés étrangères ; planificateur de requêtes	non, requêtes prédéfinies	SQL	Utilisation de formulaires ; navigation ; exécution de programmes ; affichage graphiques interactifs ; sauvegarde des résultats dans des fichiers en local ;création de liens vers sources externes
OryzaTagLine	Formalisme UML : Diagrammes de Use cases ; Diagrammes de classes ; Diagrammes de séquences	Architecture 3 tiers ; Couches métiers : perl ; Couche de persistance : BDR ; Interface : perl/HTML/Javascript	mise à jour semestrielle	Producteur de données et fournisseur données provenant d'autres sources	Matérialisée ; Spécifique d'un domaine	Création d'index uniques et non uniques sur attributs porteurs de contraintes de clés étrangères ; planificateur de requêtes	non, requêtes prédéfinies	SQL	Utilisation de formulaires ; navigation ;gestion de profils d'affichage en fonction de la confidentialité des données ; sauvegarde des résultats dans des fichiers en local ; création de liens vers sources externes
Système défini sur Le Select	Formalisme UML : Diagramme de classes pour la conception du schéma global	Modèle relationnel Architecture 3 tiers ; Couches métiers : Java/XML ; Couche de persistance : Java ; Interface : Java/XML/HTML	Directement accessibles depuis les sources	Fournisseur	Virtuelle ; Couvre plusieurs domaines	Estimation du temps de réponse d'une source par les wrappers planificateur de requêtes	oui, requêtes libres	de type SQL + Exécution de traitements + fonctions de traitements pour des chaînes de caractères	Interface de formulation de requêtes ; navigation
Système défini sur les services Web	Formalisme UML : Diagramme de séquences pour la conception des services et leurs enchaînements ; Diagramme de classe pour définir les data types	Architecture orientée-Services ; Couches métiers : Java/perl ; Interface : perl/javascript/AJAX	Directement accessibles depuis les sources	Fournisseur	Virtuelle ; Couvre plusieurs domaines	API d'enchaînement parallélisation d'appel de services	non, requêtes élémentaires prédéfinies mais possibilité de composer des requêtes	dépendante du langage de la source sous-jacentes ; dépendante des API de traitement et d'enchaînement de services	interface de manipulation de workflows de service web ; enregistrement des requêtes ; sauvegarde de résultats en ligne ;compilation de résultats en ligne ; graphiques interactifs ; création de liens vers sources externes

FIG. 7.1 – Tableau récapitulant les caractéristiques des systèmes d'intégration réalisés

7.2 Discussion

Ce travail de thèse a porté sur le partage, et au delà du partage, sur l'intégration de données et de traitements afin de permettre l'expression de requêtes complexes portant sur des sources de données distribuées potentiellement hétérogènes, ou encore, afin de permettre l'exécution de traitements complexes sur ces mêmes données.
Il s'agissait également de s'inscrire dans une démarche expérimentale que l'on pourrait qualifier de "in silico" en génomique végétale, puisque l'objectif premier était de proposer des solutions informatiques œuvrant pour une meilleure mise en commun et de fait, une meilleure utilisation des systèmes, des données et des traitements par les biologistes.

Les besoins sont particulièrement criants en génomique comme d'ailleurs en post-génomique. L'essor des biotechnologies a entraîné une production massive des données qui s'est accompagnée dans un premier temps de la définition et de la maintenance de multiples sources de données publiques, disponibles depuis le Web.
Dans un second temps, des systèmes d'intégration, relevant de l'intégration dite légère, construits sur la notion de références croisées, ont donné la possibilité aux biologistes de naviguer de source en source. L'intégration relève alors plus du processus mental du biologiste qui va croiser et interpréter les données consultées que du système lui même, qui dans ce contexte, est plutôt facilitateur de la démarche d'intégration.
Dans un troisième temps, des entrepôts de données et des systèmes de médiation ont été mis en place. Force est de constater que ces systèmes ne sont couronnés de succès que lorsqu'ils sont définis pour une communauté réduite et aux besoins bien identifiés.
Enfin sur les dernières années, de nombreuses équipes de bioinformatique se sont orientées vers la construction de systèmes d'intégration supportés par des services Web. Leur grand intérêt est notamment de pouvoir aussi bien partager et intégrer données que traitements.

Sur la base de ces constatations, nous nous sommes intéressés aux architectures de médiation ainsi qu'aux technologies issues du Web sémantique (typiquement les services Web) afin d'en dégager quelques éléments de solution de nature méthodologique.
Dans cette optique, nous avons abordé de manière concrète l'intégration de sources de données complémentaires, développées par nos soins (Oryza Tag Line et OryGenesDB) et qui ont le mérite d'être exploitées régulièrement par les biologistes du CIRAD.

Une première approche a été de mutualiser ces sources de données au sein d'une architecture de médiation nommée le Select (INRIA). Le Select a été défini tout spécifiquement pour les communautés scientifiques qui sont réputées pour manipuler des données complexes et volumineuses et pour avoir de forts besoins en terme de ressources de calcul. De cette première expérience, nous avons tiré différents enseignements en termes de mécanismes mis en place pour la consultation et le traitement de l'information possiblement hétérogène, ou encore en termes d'autonomie laissée à chacune des sources de données. Les aspects sécurité et fiabilité des données, comme les aspects optimisation de requêtes sont également très importants à nos yeux. Les médiateurs (notamment le Select) assure la protection des données (on ne peut publier que ce que l'on souhaite) et couvre l'optimisation des requêtes celle-ci étant un facteur de notoriété de ces logiciels. La fiabilité des données est un problème en soi, et dépasse le cadre de nos propositions.

Une deuxième expérience a été menée au travers d'une application dédiée, supportée par des services Web et notamment l'API BioMoby. Les éléments de réponse portent alors sur les délais d'exécution de la requête et/ou de traitement ainsi que sur la restitution graphique des résultats (synthétique et détaillée).

Par la suite, avant de tirer les enseignements en terme de " guide, conseils ", nous présentons tout d'abord une synthèse des démarches et travaux réalisés en mettant en exergue les caractéristiques conceptuelles et techniques des solutions mises en œuvre. Le fruit de la réflexion portera ensuite sur des lignes de conduite claires que l'expérience permet de dégager quant à l'exploitation optimales des ressources partagées. Les points encore en suspens et les perspectives ouvertes seront ensuite déclinées.

7.2.1 Expérimentation menée au travers de l'architecture de médiation Le Select

Le choix d'une approche de médiation est avant tout dictée par une volonté de flexibilité, de modularité et d'extensibilité au niveau de l'architecture générale du système ainsi que par une volonté d'autonomie en ce qui concerne les sources ou encore les usagers. Un tel choix permet de s'affranchir des difficultés généralement rencontrées dans le contexte des entrepôts de données telles que la lourdeur de la gestion régulière des mises à jour entre les sources de données locales et l'entrepôt. Les systèmes de médiation doivent cependant être à même de rendre compte de l'évolution des schémas des sources de données locales. De manière générale, un langage dit pivot va assurer l'unification des modèles de données des sources. Différents adaptateurs vont alors assurer le travail de traduction entre le format d'origine de la source et le format pivot de l'architecture de médiation.

Le choix du langage pivot s'avère essentiel ; il se doit d'être suffisamment expressif afin de rendre compte de toute la complexité des structures biologiques. Il se doit également d'être facile et performant à l'usage, que cela soit pour établir les règles de traduction au niveau des adaptateurs ou que cela soit pour consulter, mettre à jour ou encore contrôler les données. Dans ce cadre, Le Select est supporté par le modèle relationnel et par continuité par un langage très approchant de SQL ; et bénéfie donc de toutes les avancées réalisées dans le "monde" des bases de données relationnelles. En outre, bon nombre de bases de données en biologie moléculaire sont proposées au travers d'une vision relationnelle, ce qui facilite donc grandement une approche de médiation menée sur la base d'une modèle relationnel pivot. Le modèle relationnel possède cependant certaines limites en terme de modélisation, si l'on s'en réfère à la première forme normale ou encore aux mécanismes d'héritage et d'agrégation. D'autres langages comme CPL (Collection Programming Language) ou OQL (Object Query Language) ont été utilisés successivement comme langages pivots dans le système de médiation K2/Kleisli [DOTW97] et s'avèrent mieux adaptés pour capturer la richesse de représentation des objets biologiques (possibilité de définir des types complexes et de représenter les objets biologiques à différents niveaux de granularité). Des systèmes de médiation comme XQuare Fusion ou XLive [xli05] ont, pour leur part, adopté les langages XML Schéma et XQuery comme langages fondateurs. Dans ce sens, tout l'existant autour du langage XML et de ses extensions (typiquement XML Schéma) vont pouvoir être mis à profit. Il semble judicieux de penser que le choix du langage pivot passe donc par la considération de deux critères essentiels : l'expressivité du langage et sa

facilité d'appropriation. Il faut donc tenter de trouver le bon compromis entre ces deux critères.

Les sciences du vivant sont des sciences en perpétuelle évolution, les techniques évoluent sans cesse, les données sont multiples et complexes. Le travail de maintenance évolutive au sein des sources de données biologiques, demeure donc un travail de tous les instants. Tout nous amène donc à penser que les approches d'intégration, qui privilégient l'autonomie des sources, sont les bonnes approches. L'autonomie peut également être pensée au niveau des architectures de médiation comme au niveau des usagers. L'architecture de Le Select permet la coopération de plusieurs serveurs distribués dans le partage de données et de traitements, une approche similaire à l'architecture du système TSIMMIS [CGMH+94] qui elle-même est directement inspiré de l'architecture DARPA I3 proposé par Wiederhold [Wie92]. Les différents médiateurs impliqués dans le système, intègrent les sources, de manière décentralisée. Cette disposition possède différents avantages : le travail d'intégration comme de maintenance est partagé entre les différents médiateurs et surtout chaque médiateur peut faire preuve, dans une certaine limite, d'indépendance en proposant des points de vue originaux sur les données (au travers de mécanismes de vues par exemple) ou des outils de traitement spécifiques. Cette approche peut permettre de laisser s'exprimer toutes les sensibilités en terme de "points de vues" sur les données et les traitements à intégrer . De cette manière, les biologistes peuvent accéder à des serveurs qui contiennent une information plus riche n'ayant pas subi la vision centralisatrice et unique d'un médiateur unique et choisir librement les serveurs de médiation qui servent le mieux leurs besoins.

Le critère de modularité se révèle également un critère essentiel. Il s'avère prépondérant dans la capacité qu'aura le système à répondre aux questions biologiques posées. Un premier travail sera de s'assurer de la pertinence des sources et/ou des outils à intégrer. Il semble indispensable de n'intégrer que ce qui se révèle nécessaire d'intégrer dans le cadre de la problématique biologique considérée. Il peut ainsi être d'importance de distinguer les sources de données qui seront chevauchantes des sources de données qui seront complémentaires. Des informations additionnelles garantissant du niveau de qualité des données (indicateurs de qualité) provenant de chacune des sources peuvent également être d'utilité dans la sélection de sources realisées. De la même manière, les sources qui se verront décrites par un lot de méta-données dans un format normalisé apporteront un plus dans la démarche d'intégration. Ce lot de méta-données pourra être exploité au niveau du schéma unifié attaché au système de médiation.

La définition d'un schéma de médiation est laissé à la discrétion du concepteur dans le cadre d'une architecture de médiation telle que Le Select. Nous préconisons cependant de le construire afin d'expliciter véritablement la démarche d'intégration. Le schéma de médiation va ainsi unifier l'accès aux sources au travers du médiateur en faisant l'union des schémas locaux et en posant les correspondances qui se révèlent nécessaires entre les schémas des sources. Il définit en outre un cadre homogène et transparent pour l'exécution des requêtes.
La conception d'un tel schéma nécessite en premier lieu une étude approfondie du schéma des sources. Des différentes approches couramment répandues, nous avons privilégié l'approche Global As View (GAV) qui est mieux adaptée aux éventuels ajouts de nouvelles sources. Cette approche est toutefois moins performante que l'approche Local As View (LAV) pour ce qui concerne la construction et/ou l'optimisation des requêtes depuis le médiateur. Le choix de l'approche GAV s'effectue en général en fonction du ratio sources complémentaires / sources

chevauchantes. Plus le ratio est fort et donc plus les sources sont complémentaires et plus la construction des requêtes va se révéler simplifiée. Le médiateur sera à même de décomposer plus facilement la requête globale en sous requêtes qui chacune vont correspondre à une source donnée.
D'un point de vue pratique, pour réaliser le schéma de médiation, nous avons identifié les correspondances au niveau des sources complémentaires et construit un modèle commun pour ce qui concerne les sources chevauchantes. Ce travail est en général réalisé manuellement, il peut toutefois être facilité par la description des sources au travers de lots de méta-données et par l'existence d'une ontologie ou de plusieurs ontologies du domaine considéré. Certains systèmes de médiation vont à l'image de TAMBIS [SBB+00] utiliser des modules ontologiques pour faciliter l'élaboration des requêtes. Pour ce qui nous concerne, nous nous sommes dotés d'une ontologie décrivant les concepts de plante, de gène ou de trait fonctionnel caractérisant les plantes, qui nous a ensuite facilité le travail de réalisation du schéma unifié. Un nouveau choix est également à faire en terme du langage de représentation de connaissances qui va servir à définir puis à exploiter l'ontologie. Il va s'agir de s'assurer de la qualité de l'ontologie au travers de mécanismes de raisonnement qui vont permettre de détecter les possibles inconsistances (classification, satisfiabilité) ou encore de faciliter les étapes dites de mapping en faisant jouer des algorithmes de comparaison et de réconciliation d'ontologies (par exemple l'algorithme Prompt sous Protégé) qui vont mettre en valeur les points sémantiquement conflictuels.

Au sein de l'architecture de médiation, les adaptateurs vont rendre les correspondances entre schémas effectives, en fonction de l'approche suivie (GAV, LAV, GLAV). Le Select, dans ce cadre, met à profit les mécanismes de vue très largement exploités au sein des bases de données relationnelles pour tout ce qui concerne la sécurité, la performance mais aussi la multiplicité de la vision externe des données destinée aux usagers finaux. Le langage de définition des vues présent dans Le Select permet de manipuler les différents opérateurs spécifiques du relationnel ou ensemblistes comme l'union, pour restructurer les sources. Le Select ajoute à ce mécanisme de vues un ensemble de fonctions qui peut être étendu en fonction des besoins qui vont permettre d'opérer toutes sortes de transformations comme des transformations sur les types de données ou encore des transformations de format (image, texte, etc). Nous avons vu, dans les chapitres précédents, l'importance et le nombre des formats partagés disponibles pour la communauté des biologistes. De telles fonctionnalités de conversion de formats seront donc particulièrement appréciées par les usagers.

Ce dispositif dans son ensemble va nous permettre de prendre en charge toutes les opérations de correspondances et de transformations mais aussi œuvrer dans le sens de la sécurité. Par exemple, une vue sur les sources peut être générée selon le profil de connexion de l'usager, l'idée est de répondre de manière adaptée aux besoins des usagers mais également de protéger les données confidentielles.

Il s'agit maintenant de s'assurer de la robustesse et des performances des systèmes de médiation. L'accès aux données doit se faire dans une échelle de temps qui doit rester convenable aux yeux des usagers. Au niveau de l'architecture de médiation de Le Select, un gestionnaire de requêtes se charge de l'analyse, de l'optimisation et de l'exécution des requêtes. La phase d'analyse retourne les sources potentiellement capables de satisfaire la requête. La phase d'optimisation retient les sources les mieux à même de servir la requête après consultation de méta-données portant sur les sources (disponibilité de la source, temps de réponse estimé, etc.). Karp

[Kar95] puis Markowitz [MCKS97] ont été les premiers à souligner l'importance de la description des sources biologiques par des lots de méta-données. A l'image des médiateurs Garlic et Kleisli, (Garlic [RAH+96], Kleisli [DOTW97]), Le Select utilise donc des métadonnées pour décrire au mieux les sources. Les adaptateurs seront en charge de la gestion de ces méta-données qui peuvent être de différentes manières, par le gestionnaire de requêtes lors de la phase d'optimisation mais par les usagers qui vont ainsi pouvoir juger de la pertinence des sources à intégrer ou encore de la qualité des données. La phase d'optimisation comprend également un plan d'exécution de requêtes qui va permettre de diviser la requête en sous-requêtes qui vont ensuite être dirigées vers les adaptateurs appropriés et également de choisir les opérations à réaliser en premier lieu.

Les systèmes de médiation doivent également offrir des interfaces utilisateurs appropriées aux divers besoins des usagers qui vont se révéler multiples. En d'autres termes, les scientifiques doivent pouvoir interagir avec le médiateur de différentes manières : au travers de programmes externes imbriquant des requêtes (par exemple au travers des APIs JDBC, ODBC, ou JDO), au travers de protocoles de transfert comme ftp, au travers d'interfaces Web pour interagir visuellement avec les sources et les données, au travers directement de l'interface Web proposée de manière standard par le système médiateur. Cette dernière façon nécessite quelques compétences à composer des requêtes dans le langage du médiateur même si elle permet toute latitude pour l'expression de requêtes complexes. Nous pensons qu'un système de médiation doit offrir à ces usagers finaux, plusieurs solutions en terme d'interaction avec le système afin qu'il soit utilisé par le plus grand nombre.

7.2.2 Intégration de sources de données par le biais de services web

Le choix d'une d'intégration de sources de données par le biais de services web permet aussi de rendre accessible des données et des traitements à travers des systèmes hétérogènes et distribués (cf chapitre 6).

L'utilisation d'un annuaire de services n'est pas obligatoire, mais il permet néanmoins de centraliser le référencement des services web. En outre, l'enregistrement dans un annuaire central, ouvert et public, tel que celui du projet BioMoby permet de partager et mutualiser l'ensemble des web services développés dans le cadre d'une communauté internationale, conséquente, de bioinformaticiens. De nombreux projets du domaine de la biologie végétale développent et enregistrent des services web sur l'annuaire central BioMoby. Nous pouvons citer parmi les plus importants, le Consortium Planet qui intègre les ressources génomiques de l'espèce modèle A. thaliana, ainsi que le Generation Challenge Programme qui centralise les informations biologiques liées à la résistance des plantes tropicales aux stress abiotiques, tels que la sécheresse. Le développement de web services par l'intermédiaire de l'API BioMOBY favorise ainsi la diffusion rapide de données génomiques à une large communauté de biologistes et de bioinformaticiens.

L'enregistrement des services web sur l'annuaire central BioMOBY (Central Registry) nécessite la définition préalable des types de données ou "data types". Les data types (par exemple integer, float, string) représentent le type de contenu que peuvent prendre les données en entrée et sortie des services web. Ces data types sont structurés dans une ontologie écrite en RDF, qui

contient des relations de spécialisation "ISA" et des relations de composition "HAS et HASA". Les data types définis peuvent être complexes : le type CommentedDNASequence est défini à partir d'une spécialisation du type DNASequence et d'une composition du type Description(String). Nous avons cependant constaté que cette ontologie n'est pas correctement structurée car de nombreux types sont organisés sur un premier niveau de spécialisation. En outre, de nombreuses redondances existent parmi les data types. En effet les noms des types sont souvent préfixés pour garantir leurs unicité, malgré d'évidentes similarités ; ainsi TropGene_accession, IMGA_Accession, GCP_Identifier correspondent, en réalité, à des objets de même type. Ces incohérences dans la structure de l'ontologie des objets BioMoby ont pour conséquence de limiter la découverte automatique de services web. La connexion efficace de services web susceptibles d'être complémentaires nécessite encore l'intervention manuelle de l'utilisateur. Cette étape de définition correcte des data types est donc primordiale pour le référencement des services ainsi que leurs enchaînements. Dans ce contexte, la réutilisation de types existants est fortement recommandée lorsque les conditions s'y prêtent, notamment lorsque la description des types est déjà suffisante.

A l'image de l'intégration navigationnelle ou de l'exécution de pipeline d'analyses, les services web ont besoins d'être enchaînés pour répondre à des questions biologiques complexes. Or l'utilisation de logiciels de composition de workflows, tels que Taverna, est généralement beaucoup plus accessible à des utilisateurs experts comme des développeurs ou des bioinformaticiens. Nous avons donc développé un nouveau gestionnaire de workflows, plus adapté aux utilisateurs biologistes, qui permet de proposer une liste de workflows disponibles ou qui sélectionne les workflows adaptés aux données soumises en entrée.

Ce type d'application doit tenir compte des faiblesses des web services BioMOBY et doit s'appuyer sur le modèle relationnel pour la manipulation de données. Elle doit donc développer des fonctions permettant (i) d'éliminer la redondance (équivalent d'un distinct en SQL), (ii) de faire une différence, (iii) de cumuler des conditions (OR ou AND). Par ailleurs, un effort doit être porté sur l'optimisation des temps d'exécution pour un workflow. Dans ce domaine, nous nous sommes orientés vers la parallélisation des appels de services. Les workflows étant conçus préalablement, il est facile de déterminer quels services peuvent être appelés simultanément à partir des données soumises en entrée. Toujours au niveau de l'optimisation, il est important de distinguer dans leurs utilisations, des services appelés une fois, de ceux qui sont appelés plusieurs fois dans un même processus. Dans ce dernier cas, il est préférable de modifier les services pour qu'ils gèrent des collections de data types en entrée ce qui permet de n'utiliser le service qu'une seule fois et donc de supprimer les multiples temps d'appels vers l'annuaire central BioMoby.

Le développement d'interfaces dans ce type d'application joue un rôle important. Tout d'abords au niveau de la gestion des recherches. Par exemple, les biologistes doivent pouvoir sauvegarder plusieurs recherches ainsi que les résultats correspondants. Afin de pouvoir affiner leurs recherches et trier les résultats, des fonctions d'édition doivent être mises en place. En effet, les chercheurs compilent fréquemment des résultats nettoyés issues de plusieurs recherches. La notion de partage est également introduite à ce niveau, puisque ces résultats peuvent être partagés au sein d'un groupe. Intervient alors, la question de la gestion des groupes. En effet, comme c'est le cas pour le partage des données expérimentales, les scientifiques désirent

partager les résultats selon différents niveaux de confidentialité. Enfin, pour compléter les fonctionnalités de ces interfaces de gestion, la mise en place de systèmes d'alertes permet de garder une veille sur les recherches sauvegardées. Des alertes peuvent être placées pour les mises à jour des sources impliquées dans les workflows sauvegardés mais aussi sur l'ajout de nouvelles sources pouvant compléter l'information qui est recherchée.
Au niveau de la gestion des résultats, les outils de visualisation sont à prendre en considération. Lorsque les workflows génèrent des listes importantes de résultats, ils doivent dans un premier temps être présentés de manière synthétique (souvent une information booléenne pour les données obtenues est suffisante) afin que les scientifiques puissent réaliser des tris. Dans un deuxième temps, les résultats peuvent être affichés de manière détaillée (c'est à dire avec toutes les données obtenues par les services). Dans ce cas, l'interface doit pouvoir construire à partir des résultats, des liens vers des sources externes ce qui permet d'augmenter l'information intégrée par les services. Par exemple le numéro d'accession Genbank permet de construire un lien vers la source Genbank d'Entrez qui détaille les caractéristiques de la séquence nucléique.

7.2.3 Perspectives

A l'issue des différentes expériences menées tout au long de la thèse , nous pouvons dire que nous avons atteint les objectifs que nous nous étions fixés en terme de mutualisation et de partage de données dans le domaine de la génomique fonctionnelle végétale. Nous avons ainsi développé de manière autonome, puis rendu interopérables les systèmes OryzaTagLine et OryGenesDB tout en gardant intact leur degré d'indépendance. Nous avons également élargi notre cadre d'étude en intégrant dans notre approche d'autres sources de données (TIGR Gene Indices par exemple) et en nous appuyant sur les vocabulaires contrôlés Gene Ontology et Plant Ontology. Nos points de satisfaction principaux portent d'une part sur l'écriture de requêtes et de vues qui nous ont permis de répondre à des questions biologiques soulevant des aspects de génétique directe et indirecte, et d'autre part sur l'exploitation de données confidentielles, car non publiées, en toute sécurité, en créant différents profils utilisateurs et en leur attribuant des droits et des limitations de droit. Nous avons enfin pris en compte les besoins de la communauté utilisateur en construisant différents types d'interface en fonction des services à rendre (plus ou moins automatisés) et des compétences en informatique de chacun.

Les perspectives et les défis à relever restent nombreux. Nous pouvons distinguer une première ligne de perspectives liée à des considérations et verrous techniques identifiés :

- aller plus loin dans les mécanismes d'optimisation afin d'améliorer les performances en terme d'accès aux données. Pour ce faire, un travail plus approfondi sur les méta-données doit être mené. En effet, nous pensons qu'une exploitation parfaitement orchestrée de différents lots de méta-données peut rendre plus efficace le choix des sources à exploiter, l'écriture de la requête ou de la chaîne d'exécution du service, de la décomposition de la requête en sous-requêtes ou de la décomposition du service en sous-services et peut également permettre d'avoir une meilleure estimation de la validité et de la qualité du service ou du résultat rendu.
- nous intéresser aux activités de mise à jour dans les bases de données sous-jacentes. En effet, pour l'instant, nous avons surtout travaillé sur tout ce qui relevait de la consultation de données et essentiellement considéré des transactions en mode lecture et donc avec des verrouillages de données non bloquants. Il nous faut maintenant expérimenter

les architectures de médiation dans le contexte des transactions en mode lecture/écriture dans des environnements distribués et multi-concurrents. Outre les problèmes habituels bien identifiés par la communauté travaillant sur les transactions distribuées, se pose le problème de la notion de propriété et de propriétaire de la donnée biologique qui est un problème ouvert dans la communauté biologique.

– aller plus en avant dans l'exploitation uniformisée des ressources, que ce soit des ressources de type source de données ou des ressources de type outil de traitement. Un des points forts de Le Select est de proposer un accès uniformisé à tout type de ressource au travers de l'opérateur de "bind join" et nous n'avons qu'insuffisamment exploité ce point. Cela prend cependant tout son sens en biologie car certaines ressources, comme l'outil de recherche de similarité Blast, peuvent relever des deux, à la fois de la source de données (banque de confrontation qui correspond à des fichiers de séquence indexés) et de l'outil d'analyse.

Au-delà, des problèmes plus généraux et non forcément dédiés à la génomique fonctionnelle se dégagent :

– comment donner aux communautés la possibilité de partager des connaissances : certes nous avons noté l'importance des ontologies dans l'intégration et la médiation mais nous avons aussi remarqué que ces constructions sont souvent mal abouties.
– L'importance de la variété des points de vue devrait, à nos yeux, être partie prenante directe dans la construction des ontologies et cela constitue une réelle perspective il faut aller vers des constructions négociées intégrant les points de vue.
– Nous avons aussi signalé, l'évolutivité des données, certes les évolutions de schémas ont été largement abordées mais l'impact de ces évolutions dans les systèmes intégrés ou médiés est loin d'être pris en compte, cela aussi est une réelle perspective.

Quatrième partie

Annexes

Annexe A

Exemple de client d'appel de services web en java utilisant une connexion JDBC

```
package com.orygenesdb.services;

import fr.cirad.orygenesdb.*;

import java.sql.Connection; import java.sql.DriverManager; import
java.sql.ResultSet; import java.sql.SQLException; import
java.sql.Statement; import java.util.ArrayList; import
java.util.regex.Matcher; import java.util.regex.Pattern;

import org.biomoby.shared.MobyException; import
org.biomoby.shared.datatypes.GCP_Feature; import
org.biomoby.shared.datatypes.GCP_SimpleIdentifier; import
org.biomoby.shared.datatypes.GCP_Value; import
org.biomoby.shared.parser.MobyJob; import
org.biomoby.shared.parser.MobyPackage;

public class GetGeneInfoByGeneIdImpl extends getGeneInfoByGeneIdSkel
{
  public void processIt(MobyJob request, MobyJob response, MobyPackage context)
   throws MobyException
  {

    // Definition du pilote utilise
    String driver = "com.mysql.jdbc.Driver";

    // Definition des parametres de connexion a la base de donnees
    String url = "jdbc:mysql://[URL\_BD]ORYGENESDB\_PUBLIC";
    String url2 =  "jdbc:mysql://[URL\_BD]ARABIDOPSIS";
    String login = "[LOGIN]";
    String password = "[PASSWORD]";

    Connection connectiondb = null;
```

```
    ArrayList<GCP\_Feature> resulta_temp = new ArrayList<GCP\_Feature> ();

    /**
     * Recuperation de l'objet BioMoby de type (datatype)
     * GCP_SimpleIdentifier par la methode
     * get_gene_idSet defini dans GetGeneInfoByGeneId
     */
    GCP_SimpleIdentifier [] identifier = get_gene_idSet(request);

    // Gestion d'exception : verifie si "get_gene_idSet" a bien pris
    // l'objet BioMoby
    if (identifier ==null)
    {
      throw new MobyException("Please enter a Gene_id (ex : Os01g10900 or
      Os03g51030.1)");
}
    // nombre d'élément du tableau
    int nb_max = identifier.length;

    String gene_id[] = new String [nb_max];


    for (int i=0; i<nb_max; i++){
      if (identifier[i].getMoby_the_name()==null ||
          (identifier[i].getMoby_the_name().getValue()).equals(""))
      {
        throw new MobyException("The gene Id is empty. Please enter a
        Gene_id (ex : Os01g10900 or Os03g51030.1)");
      }

      // Gestion exception concernant "namespace"
      if ((!identifier[i].getNamespace().equals("OryGenesDB_Oryza")) &&
          (!identifier[i].getNamespace().equals("OryGenesDB_Arabidopsis")))
      {
        throw new MobyException("Namespace should be OryGenesDB_Oryza or
        OryGenesDB_Arabidopsis");
      }

      gene_id[i] = identifier[i].getMoby_the_name().getValue();
    }


    /**
     * Connection a la base et requete
     */
    try
```

```
{
  // Charger le pilote défini
  Class.forName(driver);

  // Redirection sur une base de donnees en fonction du namespace en
  // input
  // Connection a cette base
  if (identifier[0].getNamespace().equals("OryGenesDB_Oryza"))
    connectiondb = DriverManager.getConnection(url,login,password);
  if (identifier[0].getNamespace().equals("OryGenesDB_Arabidopsis"))
    connectiondb = DriverManager.getConnection(url2,login,password);

  // creation d'une instruction SQL sur la base
  Statement statement = connectiondb.createStatement();

  String query ="";

  for (int i=0; i<nb_max; i++){

    /**
     * Input Parser (use of Pattern Matching to orientate the SQL
     * request)
     */
    Pattern pattern = Pattern.compile("\\.");

    Matcher matcher = pattern.matcher(gene_id[i]);

    boolean pub_locus_model = matcher.find();
    // retourne true si l'input contient une notation pointé
    // si (true) => input est de type "pub_locus_model"
    // si (false) => input est de type "pub_locus_tu"

    // Gestion exception concernant "namespace" et le nom correct du
    // gene Id en input

    Pattern pattern2 = Pattern.compile("O");
    Matcher matcher2 = pattern2.matcher(gene_id[i]);
    boolean gene_id_os = matcher2.find();

    Pattern pattern3 = Pattern.compile("A");
    Matcher matcher3 = pattern3.matcher(gene_id[i]);
    boolean gene_id_at = matcher3.find();

    if (identifier[i].getNamespace().equals("OryGenesDB_Oryza") &&
        (!gene_id_os))
    {
```

```
    throw new MobyException("Bad gene Id for namespace
    OryGenesDB_Oryza. Please enter a valid Gene_id (ex : Os01g10900
    or Os03g51030.1)");
  }

  if (identifier[i].getNamespace().equals("OryGenesDB_Arabidopsis")
     && (!gene_id_at))
  {
    throw new MobyException("Bad gene Id for namespace
    OryGenesDB_Arabidopsis. Please enter a valid Gene_id (ex :
    AT01g10900)");
  }

  // CAS 1 : gene_id de la forme pub_locus_tu
  if (!pub_locus_model)
    query = "select distinct t.pseudochromosome_id, g.start_model,
    g.end_model, t.strand_tu, g.pub_locus_model, t.com_name,
    t.feat_name_tu, t.gene_sym, g.feat_name_model from
    TRANSCRIPT_UNIT t, GENE_MODEL g where t.tu_id=g.tu_id AND
    (t.pub_locus_tu='"+gene_id[i]+"') order by
    t.pseudochromosome_id,g.pub_locus_model";
  // CAS 2 : gene_id de la forme pub_locus_model
  if (pub_locus_model)
    query = "select distinct t.pseudochromosome_id, g.start_model,
    g.end_model, t.strand_tu, g.pub_locus_model, t.com_name,
    t.feat_name_tu, t.gene_sym, g.feat_name_model from
    TRANSCRIPT_UNIT t, GENE_MODEL g where t.tu_id=g.tu_id AND
    (g.pub_locus_model='"+gene_id[i]+"') order by
    t.pseudochromosome_id,g.pub_locus_model";

  // Execution de la requete SQL en fonction de l'input
  ResultSet results = statement.executeQuery(query);

  /**
   * Gestion des resultats
   */
          while(results.next())
          {
            // creation d'un nouvel objet avec ID : g.pub_locus_model
            GCP_Feature features  = new GCP_Feature();
            features.setId(results.getString(5));
              features.setNamespace(identifier[i].getNamespace());

            GCP_Value features1  = new GCP_Value();
            features1.setId("pseudochromosome_id");
            //features1.setNamespace(identifier[i].getNamespace());
```

```
features1.set_the_name(results.getString(1));

GCP_Value features2  = new GCP_Value();
features2.setId("start_model");
//features2.setNamespace(identifier[i].getNamespace());
features2.set_the_name(results.getString(2));

GCP_Value features3  = new GCP_Value();
features3.setId("end_model");
//features3.setNamespace(identifier[i].getNamespace());
features3.set_the_name(results.getString(3));

GCP_Value features4  = new GCP_Value();
features4.setId("strand_tu");
//features4.setNamespace(identifier[i].getNamespace());
features4.set_the_name(results.getString(4));

GCP_Value features5  = new GCP_Value();
features5.setId("com_name");
//features5.setNamespace(identifier[i].getNamespace());
features5.set_the_name(results.getString(6));

GCP_Value features6  = new GCP_Value();
features6.setId("feat_name_tu");
//features6.setNamespace(identifier[i].getNamespace());
features6.set_the_name(results.getString(7));

GCP_Value features7  = new GCP_Value();
features7.setId("gene_sym");
//features7.setNamespace(identifier[i].getNamespace());
features7.set_the_name(results.getString(8));

GCP_Value features8  = new GCP_Value();
features8.setId("feat_name_model");
//features8.setNamespace(identifier[i].getNamespace());
features8.set_the_name(results.getString(9));
features.set_values(features1);
features.set_values(features2);
features.set_values(features3);
features.set_values(features4);
features.set_values(features5);
features.set_values(features6);
features.set_values(features7);
features.set_values(features8);

resulta_temp.add(features);
```

```
                    }
        }

        // Creation d'un tableau de resultats de type GCP_Feature
        // et remplissage du tableau a partir du vecteur
        GCP_Feature tab_features2 [] = new GCP_Feature [resulta_temp.size()];
          resulta_temp.toArray(tab_features2);

          // Appel de la methode defini dans le Skel pour le renvoi des
          // resultats
          set_gene_infoSet(response,tab_features2);

    }

    catch(ClassNotFoundException cnfe){
        // System.out.println("Driver introuvable : ");
         throw new MobyException("Driver not found : please contact us");
        // cnfe.printStackTrace();
    }
    catch(SQLException sqle){
        //System.out.println("Erreur SQL : ");
        throw new MobyException("SQL error : please contact us");
        //Cf. Comment gérer les erreurs ?
    }
    catch(Exception e){
        // System.out.println("Autre erreur : ");
         throw new MobyException("Error : MobyException : please contact
                                  us");
        // e.printStackTrace();
    }

    finally
    {
        if(connectiondb!=null){try{connectiondb.close();}}
    }
  }
```

Annexe B

DTD établie pour valider le document XML final issue d'un workflow

```
<!-- "project"  -->
<!ELEMENT project (pub_locus_model)* >
<!ATTLIST project
   id                  ID     #REQUIRED
>

<!-- "pub_locus_model"  -->
<!ELEMENT pub_locus_model (gene_info,
   (est_info, fst_info, interpro, phylogenomic_info)?,
   (have_expression, have_fst, have_ipr, have_phenotype, have_est,
   have_swissprot)) >
<!ATTLIST pub_locus_model
   id                  ID     #REQUIRED
>

<!-- "gene_info"  -->
<!ELEMENT gene_info (com_name, feat_name_tu,
   gene_sym, end_model, strand_tu, start_model, feat_name_model,
   pseudochromosome_id) >
<!ATTLIST gene_info
   source              (OryGenesDB) #FIXED "OryGenesDB"
>

<!ELEMENT com_name (#PCDATA) >
<!ELEMENT feat_name_tu (#PCDATA) >
<!ELEMENT gene_sym (#PCDATA)? >
<!ELEMENT end_model (#PCDATA) >
<!ELEMENT strand_tu (+|-) >
<!ELEMENT start_model (#PCDATA) >
<!ELEMENT feat_name_model (#PCDATA) >
<!ELEMENT pseudochromosome_id (#PCDATA) >
```

```
<!-- "est_info"  -->
<!ELEMENT est_info (est_id)* >
<!ATTLIST est_info
   source              (OryGenesDB) #FIXED "OryGenesDB"
>

<!-- "est_id"  -->
<!ELEMENT est_id (EST_link, EST_description,
   EST_gff_source) >
<!ATTLIST est_id
   id                  ID     #REQUIRED
>

<!ELEMENT EST_link (#PCDATA) >
<!ELEMENT EST_description (#PCDATA)?>
<!ELEMENT EST_gff_source (#PCDATA)>

<!-- "fst_info"  -->
<!ELEMENT fst_info (fst_id)* >
<fst_info source="OryGenesDB">
<!ATTLIST fst_info
   source              (OryGenesDB) #FIXED "OryGenesDB"
>

<!-- "fst_id"  -->
<!ELEMENT fst_id (location, orientation,
   germplasm, type_insert, gff_source), phenotype_info?,
   expression_info? >
<!ATTLIST fst_id
   id                  ID     #REQUIRED
>

<!ELEMENT location (#PCDATA) >
<!ELEMENT orientation (#PCDATA) >
<!ELEMENT germplasm (#PCDATA) >
<!ELEMENT type_insert (#PCDATA) >
<!ELEMENT gff_source (#PCDATA) >

<!-- "phenotype_info"  -->
<!ELEMENT phenotype_info (plant_name)* >
<!ATTLIST phenotype_info
   source              (OryzaTagLine) #FIXED "OryzaTagLine"
>

<!-- "expression_info"  -->
<!ELEMENT expression_info (plant_name)*>
```

```
<!ATTLIST expression_info
   source              (OryzaTagLine) #FIXED "OryzaTagLine"
>

<!-- "plant_name"  -->
<!ELEMENT plant_name ((name, abreviation,
   gramene_id, plant_anatomy, known_mutant, description, keywords,
   sub_class, classe, gramene_trait)|(organ, confocal,
   stain_localisation, stain_distrib, tissu, stain_observ, stain_level,
   type, presence)) >
<!ATTLIST plant_name
   id                  ID     #REQUIRED
>

<!-- "plant_name" attributs utilisés dans la partie phenotype -->
<!ELEMENT name (#PCDATA)? >
<!ELEMENT abreviation (#PCDATA) >
<!ELEMENT gramene_id (#PCDATA)? >
<!ELEMENT plant_anatomy (#PCDATA)>
<!ELEMENT known_mutant (#PCDATA) >
<!ELEMENT description (#PCDATA)>
<!ELEMENT keywords (#PCDATA)? >
<!ELEMENT sub_class (#PCDATA) >
<!ELEMENT classe (#PCDATA) >
<!ELEMENT gramene_trait (#PCDATA)? >

<!-- "plant_name" attributs utilisés dans la partie expression -->
<!ELEMENT organ (#PCDATA) >
<!ELEMENT confocal (#PCDATA) >
<!ELEMENT stain_localisation (#PCDATA)? >
<!ELEMENT stain_distrib (#PCDATA)? >
<!ELEMENT tissu (#PCDATA) >
<!ELEMENT stain_observ (#PCDATA)? >
<!ELEMENT stain_level (#PCDATA)? >
<!ELEMENT type (#PCDATA) >
<!ELEMENT presence (#PCDATA) >

<!-- "interpro"  -->
<!ELEMENT interpro (ipr_id)* >
<interpro source="GreenPhylDB">
<!ATTLIST phenotype_info
   source              (GreenPhylDB) #FIXED "GreenPhylDB"
>

<!-- "ipr_id"  -->
<!ELEMENT ipr_id (ipr_end, ipr_start, ipr_score,
```

```
   type, description) >
<!ATTLIST ipr_id
   id                  ID     #REQUIRED
>

<!ELEMENT ipr_end (#PCDATA) >
<!ELEMENT ipr_start (#PCDATA) >
<!ELEMENT ipr_score (#PCDATA) >
<!ELEMENT type (#PCDATA) >
<!ELEMENT description (#PCDATA) >

<!-- "phylogenomic_info"  -->
<!ELEMENT phylogenomic_info (ortholog)* >
<!ATTLIST phylogenomic_info
   source              (GreenPhylDB) #FIXED "GreenPhylDB"
>

<!-- "ortholog"  -->
<!ELEMENT ortholog (s, D, n, o)|(D,p) >
<!ATTLIST ortholog
   id                  ID     #REQUIRED
>

<!-- "ortholog" attributs utilisés dans la partie orthologie -->
<!ELEMENT s (#PCDATA) >
<!ELEMENT D (#PCDATA) >
<!ELEMENT n (#PCDATA) >
<!ELEMENT o (#PCDATA) >

<!-- "ortholog" attributs utilisés dans la partie ultra-paralogie -->
<!ELEMENT p (#PCDATA) >

<!-- "les tags suivant notifient la présence ou l'absence
d'informations, et servent à l'affichage syntaxique des résultats"
-->
<!ELEMENT have_expression #REQUIRED type(Yes|No) >
<!ELEMENT have_fst #REQUIRED type(Yes|No) >
<!ELEMENT have_ipr #REQUIRED type(Yes|No) >
<!ELEMENT have_phenotype #REQUIRED type(Yes|No) >
<!ELEMENT have_est #REQUIRED type(Yes|No) >
<!ELEMENT have_swissprot #REQUIRED type(Yes|No) >
```

Annexe C

Document XML final issue d'un workflow

```
<pub_locus_model id="Os01g10900.1">
    <gene_info source="OryGenesDB">
        <com_name>ATP binding protein, putative, expressed</com_name>
        <end_model>5816009</end_model>
        <strand_tu>-</strand_tu>
        <is_te>0</is_te>
        <start_model>5811583</start_model>
        <pseudochromosome_id>1</pseudochromosome_id>
        <pub_locus_model>Os01g10900.1</pub_locus_model>
        <name_tu>12001.t00963</name_tu>
    </gene_info>
    <phylogenomic_info source="GreenPhylDB">
        <ortholog id="At4g18640.1">
            <super-orthology>0</super-orthology>
            <distance>0.7193</distance>
            <subtree neighboring>99</subtree neighboring>
            <orthology>100</orthology>
        </ortholog>
    </phylogenomic_info>
    <interpro source="GreenPhylDB">
        <ipr_id id="IPR000719">
            <ipr_end>643</ipr_end>
            <ipr_start>403</ipr_start>
            <ipr_score>7e-72</ipr_score>
            <type>Domain</type>
            <description>Protein kinase</description>
        </ipr_id>
            <ipr_end>83</ipr_end>
            <ipr_start>42</ipr_start>
            <ipr_score>1.8e-10</ipr_score>
            <type>Domain</type>
            <description>Leucine rich repeat, N-terminal</description>
        </ipr_id>
    </interpro>
```

```
    <have_expression>0</have_expression>
    <have_fst>0</have_fst>
    <have_phenotype>0</have_phenotype>
</pub_locus_model> <pub_locus_model id="Os01g10900.2">
    <gene_info source="OryGenesDB">
        <com_name>ATP binding protein, putative, expressed</com_name>
        <end_model>5815612</end_model>
        <strand_tu>-</strand_tu>
        <is_te>0</is_te>
        <start_model>5811583</start_model>
        <pseudochromosome_id>1</pseudochromosome_id>
        <pub_locus_model>Os01g10900.2</pub_locus_model>
        <name_tu>12001.t00963</name_tu>
    </gene_info>
    <fst_info source="OryGenesDB">
        <fst_id id="TEBA3E08">
            <location>Exon</location>
            <orientation>forward</orientation>
            <germplasm>ADRA11</germplasm>
            <type_insert>Tos17</type_insert>
            <gff_source>OTL</gff_source>
            <phenotype_info source="OryzaTagLine">
                <plant_name id="ADRA11">
                    <name></name>
                    <abreviation>rc</abreviation>
                    <gramene_id>TO:0000487</gramene_id>
                    <plant_anatomy>endosperm</plant_anatomy>
                    <known_mutant>brown pericarp and seed coat</known_mutant>
                    <description>brown endosperm, glume colour is wild type</description>
                    <keywords>Brown</keywords>
                    <sub_class></sub_class>
                    <classe>morphology</classe>
                    <gramene_trait>endosperm color</gramene_trait>
                </plant_name>
            </phenotype_info>
            <expression_info source="OryzaTagLine">
                <plant_name id="ADRA11">
                    <organ>embryo</organ>
                    <confocal>0</confocal>
                    <stain_localisation></stain_localisation>
                    <stain_distrib></stain_distrib>
                    <tissu></tissu>
                    <stain_observ></stain_observ>
                    <stain_level>none</stain_level>
                    <type>GUS</type>
                    <presence>0</presence>
```

```
                    </plant_name>
                </expression_info>
            </fst_id>
        </fst_info>
        <have_expression>0</have_expression>
        <have_fst>1</have_fst>
        <have_phenotype>1</have_phenotype>
</pub_locus_model>
```

Annexe D

Glossaire

ADN complémentaire *ADNc* : Brin d'ADN copié à partir d'un brin d'ARN par la transcriptase inverse. La séquence d'ADNc est complémentaire de celle de l'ARN utilisé comme matrice.

ADN polymérase (ADN ou ARN-dépendante) : Enzyme qui allonge les brins d'ADN en y joignant des nucléotides individuels par des liaisons chimiques covalentes. L'ordre des nucléotides reste cependant dicté par celui des nucléotides complémentaires présents dans un brin d'ADN (ou d'ARN), la matrice.

ADN-T (ADN de transfert) : Fragment d'ADN encadré par les séquences bordures droite et gauche (répétition de 24 paires de bases) du plasmide Ti et Ri d'Agrobactérium qui permettent l'intégration stable de ce fragment dans le génome d'une plante.

Algorithme : Procédure de résolution d'un problème qui peut être traduit en langage informatique afin de produire "automatiquement" un résultat. Enchaînement des actions nécessaires à l'accomplissement "automatique" d'une tâche.

Allèle : Version possible d'un locus. Le locus peut être un gène ou une séquence d'ADN non codante. Lorsque, chez différents individus, différentes formes existent à un locus donné, chacune de ces formes est un allèle.

Allogame : C'est un mode de fécondation croisée chez les plantes. Dans ce cas, la fécondation a lieu avec un gamète male issu d'un individu et un gamète femelle issu d'un autre individu

API (Application Programming Interface ou Interface de programmation) : permet de définir la manière dont un composant informatique peut communiquer avec un autre. C'est donc une interface de code source fournie par un système informatique ou une bibliothèque logicielle, en vue de répondre à des requêtes pour des services qu'un programme informatique pourrait lui faire.

Applet : Programme en Java conçu pour être téléchargé via un réseau à chaque fois qu'on veut l'exécuter, en général par un navigateur web.

Autogame : C'est le mode de reproduction d'une plante. Dans ce cas, les deux gametes sont issus du même individu, c'est une autofécondation

BAC (Bacterial Artificial Chromosome) : Vecteur bactérien permettant de cloner des fragments d'ADN de l'ordre de 100 à 200 kilobases dans des cellules d'Escherichia coli. Les BAC

peuvent être manipulés comme des plasmides de trés grande taille. Banque génomique. Collection de fragments d'ADN génomique représentative de la totalité d'un génome ; ces fragments sont clonés dans des vecteurs (BAC,tec.) et propagés dans des cellules hôtes (bactéries, levures, etc.).

BBMH (Best Blast Mutual Hit) : cette méthode consiste à exécuter des blasts sur les deux protéomes. Les gènes qui sont identifiés comme orthologues auront un alignement réciproque entre les deux espèces.

BioMOBY : Projet pour la découverte, l'intégration et l'interopérabilité de services et de bases de données biologiques.

BioPerl : Ensemble de librairies écrites en langage Perl et dédiées au domaine de la bioinformatique.

BLAST : Logiciels qui permettent de comparer des séquences deux à deux et de déterminer leurs régions d'homologie.

Cals : Il s'agit de cals cellulaires, c'est à dire un amas de cellules indifférenciées. Chaque cellule du cal peut être à l'origine d'une plante.

Carte génétique : Agencement le long des chromosomes de locus dont les positions relatives sont déterminées à partir des fréquences de recombinaison entre les locus. Les distances génétiques, exprimées en centimorgan (cM), sont donc fonction des taux de recombinaison observés et dépendent ainsi de la population de cartographie utilisée.

Carte physique du génome : Reconstitution du génome par un ensemble de fragments d'ADN ordonnés les uns par rapport aux autres et positionnés le long des chromosomes au moyen de marqueurs moléculaires. Les distances entre les marqueurs sont exprimées en paires de bases (pb).

Centromère : Région spécialisée du chromosome eucaryotique requise pour la répartition des chromosomes dans les cellules filles au cours de la division cellulaire.

Client : En informatique, le mot client est employé dans le contexte du modèle client/serveur. C'est un logiciel installé sur le poste de travail qui permet d'accéder à un serveur du même type.

Colinéarité : Conservation de l'ordre des gènes le long de leur chromosome.

CVS (Concurrent Versions System) : Outil de gestion de sources multi-utilisateurs permettant de sauver et de récupérer les différentes versions de fichiers.

DAML-S (DARPA Agent Markup Language Services) : Langage de description sémantique de services web. Il permet la description, la recherche, la sélection et l'exécution d'un service web particulier mais aussi la composition de services entre eux.

DAML+OIL : Langage permettant de définir une ontologie pour un domaine particulier.

Dataflow : Application dans laquelle la modification de la valeur d'une variable entraîne automatiquement la réévaluation des variables qui en dépendent (un tableur est un exemple de dataflow). Les termes dataflow et workflow sont quelquefois assimilés.

DTD (Document Type Definition) : Document permettant de définir la structure d'un fichier XML.

EMBOSS : Ensemble de logiciels pour la biologie moléculaire.

EST (Expressed Sequence Tag ou etiquettes de séquences transcrites) : Courtes séquences de 300 à 500 nucléotides résultant du séquençage partiel de chacun des clones de banques d'ADNc. Ces séquences reflètent l'expression des gènes dans une cellule à un instant donné.

Framework : Infrastructure logicielle qui facilite la conception des applications par l'utilisation de bibliothèques ou de générateurs de programmes. En français "un cadre de développement".

FST (Flanking Sequence Tag) : Séquence d'une région d'ADN génomique adjacente à une insertion d'ADN étranger connu (ADN-T, transposons).

Gène candidat : Gène dont la fonction physiologique, la localisation et/ou le polymorphisme laissent supposer que sa variabilité joue un rôle dans une variation phénotypique.

Gène homologue : Deux gènes homologues sont issus d'un ancêtre commun.

Gène orthologue : Deux gène orthologues sont deux gènes homologues codant pour les mêmes fonctions et dérivant d'un même gène ancestral. Ils appartiennent donc à deux espèces distinctes issues d'un ancêtre commun et présentent éventuellement une position conservée dans le génome de ces deux espèces.

Gène rapporteur : Gène dont l'expression phénotypique est facilement détectable et qui peut être utilisé sous le contrôle de région promotrices pour en étudier l'activité dans l'espace ou le temps (gènes GUS, CAT, luciférase, GFP,etc.)

Génétique inverse : Elle consiste à identifier des mutations dans un gêne et à en analyser les conséquences sur le phénotype, c'est la démarche inverse de la génétique classique.

Génomique : Etude globale et systématique des génomes dans le but d'obtenir une vue générale de leur organisation et de leur fonctionnement. La génomique structurale permet de décrire l'organisation des chromosomes et de dresser l'inventaire des gènes qu'ils contiennent et la génomique fonctionnelle vise à attribuer des fonctions à ces gènes et à comprendre l'ensemble de leurs régulations et de leurs interactions.

GFF (General Feature Format) : Il s'agit d'un format de fichier semi-structuré permettant de décrire l'annotation des séquences. GFF (http ://www.sanger.ac.uk/Software/formats/GFF/) permet notamment de structurer les informations que l'on qualifie de «features» dans la description des séquences d'ADN, d'ARN et protéiques. **GFP (Green Fluorescent Protein)** : Protéine

fluorescent isolée d'une méduse (Aequorea victoria).

GUS : Gène rapporteur de la béta-glucuronidase dont l'activité peut être dosée ou visualisée *in situ*.

GRID : computing Architecture de groupes d'ordinateurs reliés entre eux par un réseau étendu pour exécuter des tâches et pour lesquels les échanges peuvent être lents.

Homologue : L'homologie est la ressemblance héritée d'un même ancêtre commun. Deux séquences sont dites homologues si elles ont un ancêtre commun.

Interopérabilité : L'interopérabilité est la capacité que possède un produit ou un système dont les interfaces sont intégralement connues à fonctionner avec d'autres produits ou systèmes existants ou futurs.

MyGrid : Projet visant à développer une application de calcul distribué dédiée à la bioinformatique. Le but est aussi de permettre la recherche de services web, et l'exécution de workflows utilisant des ressources distribués sur un réseau étendu.

Monocolylédone : Parmi les angiospermes ou plantes à fleurs, les monocotylédones comprennent des végétaux dont la plantule typique ne présente qu'un seul cotylédon sur l'embryon, qui évolue en donnant une pré-feuille.

Mutagénèse : Il existe 3 types de mutagénèse : la mutagénèse chimique (EMS), la mutagénèse physique (rayon X, neutrons, etc) et la mutagénèse insertionnelle. Alors que les deux premières provoquent des modifications de l'ADN par transformation ou délétion de bases, la mutagénèse insertionnelle introduit un élément d'ADN étranger dans le génome hôte.

Ontologie : une ontologie représente l'ensemble des connaissances d'un domaine, au travers d'une hiérarchie de concepts liés par des relations sémantiques. Une ontologie se doit d'être exploitable par un logiciel (description formelle) comme par un opérateur humain (description littéraire)

Orthologue : Gènes orthologues : gènes d'espèces différentes dont les séquences sont homologues, la divergence faisant suite à une spéciation. S'il s'agit d'une évolution après duplication au sein d'un individu on parlera de paralogue.

Parser : programme permettant d'analyser la structuce syntaxique d'un fichier texte afin d'en réaliser des traitements.

Polymorphisme génétique : Variation entre individus dans la séquence de gènes. Ces variations qui rendent compte des différents allèles dans une population sont normales et ne sont pas pathogènes.

Primer : Courte séquence nucléotidique servant de support pour une amplification d'ADN par PCR

QTL (Quantitative Trait Locus) : Locus contrôlant la variation d'un caractère quantitatif. Il s'agit en fait de la détection statistique d'un gène (ou de plusieurs gènes finement liés) dont le polymorphisme explique une partie de la variabilité phénotypique observée dans une population adéquate (lignée recombinante, F2, ..) pour un caractère quantitatif. Cette détection statistique repose sur l'analyse de marqueurs génétiques cartographiés sur la même population.

RDF (Resource Description Framework) : Modèle et description de syntaxe en vue de l'utilisation de méta données sur le web. Son objectif est de faciliter le traitement automatisé des informations contenues sur le web en permettant leur description ans ambiguïté.

service Web : Application disponible sur le web et accessible par une interface standardisée. Elle peut interagir avec d'autres services web indépendamment du système d'exploitation et des langages de programmation utilisés.

SOAP (Simple Object Access Protocol) : Protocole de communication inter-applicatif, au

dessus de HTTP, comportant un ensemble de règles pour structurer des messages (XML) et invoquer un service web.

Soaplab : Ensemble de services Web permettant l'accès à des applications, essentiel ement d'analyse de données. Intègre plus particulièrement la suite EMBOSS en proposant une définition XML des fichiers ACD. Il intègre aussi un service d'enregistrement et de recherche de services web.

Synténie : Conservation du regroupement des gènes sur un même chromosome dans certaines portions du génome d'espèces différentes.

Thread : Portion de programme pouvant s'exécuter en parallèle d'autres portions.

UDDI (Universal Description Discovery and Integration) : Norme permettant de créer et de retrouver des services web. Un annuaire UDDI est un annuaire en ligne, basé sur la norme UDDI, référençant un ensemble de services Web disponibles.

workflow : Application qui permet de séquencer des tâches suivant un modèle qui définit en particulier comment ces tâches sont synchronisées. Voir aussi dataflow.

WSDL (Web Service Description Language) : Langage basé sur XML permettant la description de l'interface d'un service web.

XML (eXtensible Markup Languages) : Format universel de stockage et d'échange de données. XML est un langage de balisage extensible, c'est-à-dire qu'il n'est pas sémantiquement figé comme HTML. Il permet de définir ses propres balises, ce qui le rend adaptable et donc à même de stocker tous types d'informations. La force de la norme XML est de séparer les données et leur présentation (le fond et la forme).

XQuery : Langage pour interroger et manipuler les données d'un document XML.

XSLT (eXtensible Style Language Transformation) : langage permettant de transformer un document XML en un nouveau document XML ayant une structure (et éventuellement une DTD) différente.

Bibliographie

[ABB+00] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology : tool for the unification of biology. the gene ontology consortium. *Nat Genet*, 25(1) :25–29, May 2000.

[ABW+04] Rolf Apweiler, Amos Bairoch, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, Maria J Martin, Darren A Natale, Claire O'Donovan, Nicole Redaschi, and Lai-Su L Yeh. Uniprot : the universal protein knowledgebase. *Nucleic Acids Res*, 32(Database issue) :D115–D119, Jan 2004.

[ABY+07] B. A. Antonio, C. R. Buell, Y. Yamazaki, I. Yap, C. Perin, and R. Bruskiewich. *Informatics Resources for Rice Functional Genomics*, chapter 14, pages 355–394. Springer, 2007.

[ACH+00] M. D. Adams, S. E. Celniker, R. A. Holt, C. A. Evans, J. D. Gocayne, and et al. The genome sequence of drosophila melanogaster. *Science*, 287(5461) :2185–2195, Mar 2000.

[AGM+90] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3) :403–410, Oct 1990.

[AMM44] O.T. Avery, C.M. MacLeod, and M. MacCarty. Studies on the chemical nature of the substance inducing transformation of the substance inducing transformation of pneumococcal types. *J. Exp. Med.*, 79 :137–158, 1944.

[ASL+05] Motoyuki Ashikari, Hitoshi Sakakibara, Shaoyang Lin, Toshio Yamamoto, Tomonori Takashi, Asuka Nishimura, Enrique R Angeles, Qian Qian, Hidemi Kitano, and Makoto Matsuoka. Cytokinin oxidase regulates rice grain production. *Science*, 309(5735) :741–745, Jul 2005.

[BAW+05] Amos Bairoch, Rolf Apweiler, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, Maria J Martin, Darren A Natale, Claire O'Donovan, Nicole Redaschi, and Lai-Su L Yeh. The universal protein resource (uniprot). *Nucleic Acids Res*, 33(Database issue) :D154–D159, Jan 2005.

[BBB+98] P. G. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, and R. Stevens. Tambis–transparent access to multiple bioinformatics information sources. *Proc Int Conf Intell Syst Mol Biol*, 6 :25–34, 1998.

[BBF+86] H. S. Bilofsky, C. Burks, J. W. Fickett, W. B. Goad, F. I. Lewitter, W. P. Rindone, C. D. Swindell, and C. S. Tung. The genbank genetic sequence databank. *Nucleic Acids Res*, 14(1) :1–4, Jan 1986.

[BCD+04] Alex Bateman, Lachlan Coin, Richard Durbin, Robert D Finn, Volker Hollich, Sam Griffiths-Jones, Ajay Khanna, Mhairi Marshall, Simon Moxon, Erik L L Sonnhammer, David J Studholme, Corin Yeats, and Sean R Eddy. The pfam protein families database. *Nucleic Acids Res*, 32(Database issue) :D138–D141, Jan 2004.

[BCM+03] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P.F. Patel-Schneider. *The Description Logic Handbook : Theory, Implementation, and Applications.* Cambridge University Press., 2003.

[BDH+95] P. Buneman, S. B. Davidson, K. Hart, C. Overton, and L. Wong. A data transformation system for biological data sources. In *Proceedings of the Twenty-first International Conference on Very Large Databases*, Zurich, Switzerland, 1995. VLDB Endowment, Saratoga, Calif.

[BDH+06] Richard Bruskiewich, Guy Davenport, Tom Hazekamp, Thomas Metz, Manuel Ruiz, Reinhard Simon, Masaru Takeya, Jennifer Lee, Martin Senger, Graham McLaren, and Theo Van Hintum. Generation challenge programme (gcp) : standards for crop data. *OMICS*, 10(2) :215–219, 2006.

[BFG+85] C. Burks, J. W. Fickett, W. B. Goad, M. Kanehisa, F. I. Lewitter, W. P. Rindone, C. D. Swindell, C. S. Tung, and H. S. Bilofsky. The genbank nucleic acid sequence database. *Comput Appl Biosci*, 1(4) :225–233, Dec 1985.

[BFH+05] Louis M T Bradbury, Timothy L Fitzgerald, Robert J Henry, Qingsheng Jin, and Daniel L E Waters. The gene for fragrance in rice. *Plant Biotechnol J*, 3(3) :363–370, May 2005.

[BGB+99] P. G. Baker, C. A. Goble, S. Bechhofer, N. W. Paton, R. Stevens, and A. Brass. An ontology for bioinformatics applications. *Bioinformatics*, 15(6) :510–520, Jun 1999.

[BH98] D. Bouchez and H. Höfte. Functional genomics in plants. *Plant Physiol*, 118(3) :725–732, Nov 1998.

[BHGS01] Sean Bechhofer, Ian Horrocks, Carole Goble, and Robert Stevens. OilEd : A reason-able ontology editor for the semantic Web. *Lecture Notes in Computer Science*, 2174 :396– ? ?, 2001.

[BKML+06] Dennis A Benson, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, and David L Wheeler. Genbank. *Nucleic Acids Res*, 34(Database issue) :D16–D20, Jan 2006.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lasilla. The semantic web. *Scientific American*, 2001.

[BLT93] M. S. Boguski, T. M. Lowe, and C. M. Tolstoshev. dbest–database for "expressed sequence tags". *Nat Genet*, 4(4) :332–333, Aug 1993.

[BS85] R. J. Brachman and J. G. Schmolze. An overview of the kl-one knowledge representation systems. *Cogn. Sci.*, 9 (2) :pp. 171–216., 1985.

[BT41] G. W. Beadle and E. L. Tatum. Genetic control of biochemical reactions in neurospora. *Proc Natl Acad Sci U S A*, 27(11) :499–506, Nov 1941.

[CBDF+06] Sarah Cohen-Boulakia, Susan Davidson, Christine Froidevaux, Zoé Lacroix, and Maria-Esther Vidal. Path-based systems to guide scientists in the maze of biological data sources. *J Bioinform Comput Biol*, 4(5) :1069–1095, Oct 2006.

[CBFP06] Sarah Cohen-Boulakia, Christine Froidevaux, and Emmanuel Pietriga. Selecting biological data sources and tools with xpr, a path language for rdf. *Pac Symp Biocomput*, pages 116–127, 2006.

[CBL+04] Evelyn Camon, Daniel Barrell, Vivian Lee, Emily Dimmer, and Rolf Apweiler. The gene ontology annotation (goa) database–an integrated resource of go annotations to the uniprot knowledgebase. *In Silico Biol*, 4(1) :5–6, 2004.

[CCH+93] M. T. Chan, H. H. Chang, S. L. Ho, W. F. Tong, and S. M. Yu. Agrobacterium-mediated production of transgenic rice plants expressing a chimeric alpha-amylase promoter/beta-glucuronidase gene. *Plant Mol Biol*, 22(3) :491–506, Jun 1993.

[CFF+06] Ron Caspi, Hartmut Foerster, Carol A Fulcher, Rebecca Hopkinson, John Ingraham, Pallavi Kaipa, Markus Krummenacker, Suzanne Paley, John Pick, Seung Y Rhee, Christophe Tissier, Peifen Zhang, and Peter D Karp. Metacyc : a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res*, 34(Database issue) :D511–D516, Jan 2006.

[CG93] F. Collins and D. Galas. A new five-year plan for the u.s. human genome project. *Science*, 262(5130) :43–46, Oct 1993.

[CG06] Sébastien Carrere and Jérôme Gouzy. Remora : a pilot in the ocean of biomoby web-services. *Bioinformatics*, 22(7) :900–901, Apr 2006.

[CGMH+94] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The Tsimmis project : Integration of heterogenous information sources. In *Proceedings of 100th Anniversary Meeting of the Information Processing Society of Japan*, pages 7–18, Tokyo, Japan, October 1994.

[Cha64] MF Chandraratna. Genetics and breeding of rice. Longmans, London UK, 1964.

[CHN+95] William F. Cody, Laura M. Haas, Wayne Niblack, Manish Arya, Michael J. Carey, Ronald Fagin, Myron Flickner, Denis Lee, Dragutin Petkovic, Peter M. Schwarz, Joachim Thomas II, Mary Tork Roth, John H. Williams, and Edward L. Wimmers. Querying multimedia data from multiple repositories by content : the garlic project. In *VDB*, pages 17–35, 1995.

[CHS+03] Peter A Covitz, Frank Hartel, Carl Schaefer, Sherri De Coronado, Gilberto Fragoso, Himanso Sahni, Scott Gustafson, and Kenneth H Buetow. caCORE : a common infrastructure for cancer informatics. *Bioinformatics*, 19(18) :2404–2412, Dec 2003.

[CMB+03] Evelyn Camon, Michele Magrane, Daniel Barrell, David Binns, Wolfgang Fleischmann, Paul Kersey, Nicola Mulder, Tom Oinn, John Maslen, Anthony Cox, and Rolf Apweiler. The gene ontology annotation (goa) project : implementation of go in swiss-prot, trembl, and interpro. *Genome Res*, 13(4) :662–672, Apr 2003.

[CMB+04] Evelyn Camon, Michele Magrane, Daniel Barrell, Vivian Lee, Emily Dimmer, John Maslen, David Binns, Nicola Harte, Rodrigo Lopez, and Rolf Apweiler. The gene ontology annotation (goa) database : sharing knowledge in uniprot with gene ontology. *Nucleic Acids Res*, 32(Database issue) :D262–D266, Jan 2004.

[CMC+02] Maria Cláudia Cavalcanti, Marta Mattoso, Maria Luiza Machado Campos, Eric Simon, and François Llirbat. An architecture for managing distributed scientific resources. In *SSDBM*, pages 47–, 2002.

[Con] International Human Genome Sequencing Consortium. initial sequencing and analysis of the human genome. *nature*, 409 :860–921.

[Con01] Gene Ontology Consortium. Creating the gene ontology resource : design and implementation. *Genome Res*, 11(8) :1425–1433, Aug 2001.

[Cou88] Brigitte Courtois. Les systemes de culture du riz pluvial. Memoires et Travaux de l'IRAT 16, 1988.

[Cou07] Brigitte Courtois. Une brève histoire de l'amélioration génétique du riz http ://tropgenedb.cirad.fr/rice/ameliorationgenetiqueriz.pdf, 2007.

[DCB+01] Susan B. Davidson, Jonathan Crabtree, Brian Brunk, Jonathan Schug, Val Tannen, Chris Overton, and Chris Stoeckert. K2kleisli and gus : Experiments in integrated access to genomic data sources. *IBM Systems Journal*, 40(2) :512–31, 2001.

[DEP72] M. O. Dayhoff, R. V. Eck, and C.M. Parck. Atlas of protein sequence and structure. *National Biomedical Research Foundation*, 5 :75–84, 1972.

[DOB95] S.B. Davidson, C. Overton, and P. Buneman. Challenges in integrating biological data sources. *J Comput Biol*, 2(4) :557–72, 1995.

[DOTW97] Susan B. Davidson, G. Christian Overton, Val Tannen, and Limsoon Wong. Biokleisli : A digital library for biomedical researchers. *Int. J. on Digital Libraries*, 1(1) :36–53, 1997.

[DRL+06] G. Droc, M. Ruiz, P. Larmande, A. Pereira, P. Piffanelli, J. B. Morel, A. Dievart, B. Courtois, E. Guiderdoni, and C. Périn. Orygenesdb : a database for rice reverse genetics. *Nucleic Acids Res*, 34(Database issue) :D736–D740, Jan 2006.

[DS04] Mike Dean and Guus Schreiber. Owl web ontology language reference. W3C Recommendation, 2004.

[EA93] T. Etzold and P. Argos. SRS–an indexing and retrieval tool for flat file data libraries. *Comput Appl Biosci*, 9(1) :49–57, Feb 1993.

[ELR01] B.A. Eckman, Z. Lacroix, and L. Raschid. Optimized seamless integration of biomolecular data. *IEEE symposium on Bio-Informatics and Biomedical Engenieering (BIBE'01), Washington DC*, pages 23–32, 2001.

[eSC98] C. elegans Sequencing Consortium. Genome sequence of the nematode c. elegans : a platform for investigating biology. *Science*, 282(5396) :2012–2018, Dec 1998.

[EUA96] T. Etzold, A. Ulyanov, and P. Argos. SRS : information retrieval system for molecular biology data banks. *Methods Enzymol*, 266 :114–28, 1996.

[FAW+95] R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bult, J. F. Tomb, B. A. Dougherty, and J. M. Merrick. Whole-genome random sequencing and assembly of haemophilus influenzae rd. *Science*, 269(5223) :496–512, Jul 1995.

[FHvH+00] Dieter Fensel, Ian Horrocks, Frank van Harmelen, Stefan Decker, Michael Erdmann, and Michel C. A. Klein. OIL in a nutshell. In *Knowledge Acquisition, Modeling and Management*, pages 1–16, 2000.

[Fra97] J.M. Franco. *Le Data Warehouse : objectifs, définitions, architectures* . Eyrolles., 1997.

[Gal07] Michael Y Galperin. The molecular biology database collection : 2007 update. *Nucleic Acids Res*, 35(Database issue) :D3–D4, Jan 2007.

[Gar08] A.E. Garrod. Inborn errors of metabolism. *Lancet*, 2 :1–7,73–79,142–148,214–220, 1908.

[GGH+03] Elisabeth Gasteiger, Alexandre Gattiker, Christine Hoogland, Ivan Ivanyi, Ron D Appel, and Amos Bairoch. Expasy : The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res*, 31(13) :3784–3788, Jul 2003.

[GHBC+02] Margarita Garcia-Hernandez, Tanya Z Berardini, Guanghong Chen, Debbie Crist, Aisling Doyle, Eva Huala, Emma Knee, Mark Lambrecht, Neil Miller, Lukas A Mueller, Suparna Mundodi, Leonore Reiser, Seung Y Rhee, Randy Scholl, Julie Tacklind, Dan C Weems, Yihe Wu, Iris Xu, Daniel Yoo, Jungwon Yoon, and Peifen Zhang. Tair : a resource for integrated arabidopsis data. *Funct Integr Genomics*, 2(6) :239–253, Nov 2002.

[GMB+05] E. Guérin, G. Marquet, Anita Burgun, O. Loréal, Laure Berti-Equille, Ulf Leser, and Fouzia Moussouni. Integrating and warehousing liver gene expression data and related biomedical resources in gedaw. In *DILS*, pages 158–174, 2005.

[Gof96] A. Goffeau. 1996 : a vintage year for yeast and yeast. *Yeast*, 12(16) :1603–1605, Dec 1996.

[GRL+02] Stephen A Goff, Darrell Ricke, Tien-Hung Lan, Gernot Presting, Ronglin Wang, and et al. A draft sequence of the rice genome (oryza sativa l. ssp. japonica). *Science*, 296(5565) :92–100, Apr 2002.

[Gru93] Thomas R. Gruber. Towards principles for the design of ontologies used for knowledge sharing in formal ontology in conceptual analysis and knowledge representation. *Kluwer Academic Publishers*, 1993.

[Gué05] Emilie Guérin. *Integration de donnees pour l'analyse de transcriptome : mise en oeuvre par l'entrepot GEDAW (Gene Expression Data Warehouse)*. PhD thesis, Univ. de Rennes 1, 2005.

[Hal01] Alon Y. Halevy. Answering queries using views : A survey. *VLDB Journal : Very Large Data Bases*, 10(4) :270–294, 2001.

[HBB+02] T. Hubbard, D. Barker, E. Birney, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down, R. Durbin, E. Eyras, J. Gilbert, M. Hammond, L. Huminiecki, A. Kasprzyk, H. Lehvaslaiho, P. Lijnzaad, C. Melsopp, E. Mongin, R. Pettett, M. Pocock, S. Potter, A. Rust, E. Schmidt, S. Searle, G. Slater, J. Smith, W. Spooner, A. Stabenau, J. Stalker, E. Stupka, A. Ureta-Vidal, I. Vastrik, and M. Clamp. The ensembl genome database project. *Nucleic Acids Res*, 30(1) :38–41, Jan 2002.

[HC86] G. H. Hamm and G. N. Cameron. The embl data library. *Nucleic Acids Res*, 14(1) :5–9, Jan 1986.

[HCI+04] M. A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, and et al. The gene ontology (go) database and informatics resource. *Nucleic Acids Res*, 32(Database issue) :D258–D261, Jan 2004.

[HDM+] Farshad Hakimpour, John Domingue, Enrico Motta, Liliana Cabral, and Yuangui Lei. Integration of owl-s into irs-iii.

[HFB+04] M. Hucka, A. Finney, B. J. Bornstein, S. M. Keating, B. E. Shapiro, J. Matthews, B. L. Kovitz, M. J. Schilstra, A. Funahashi, J. C. Doyle, and H. Kitano. Evolving a lingua franca and associated software infrastructure for computational systems biology : the systems biology markup language (sbml) project. *Syst Biol (Stevenage)*, 1(1) :41–53, Jun 2004.

[HGA+04] Hirohiko Hirochika, Emmanuel Guiderdoni, Gynheung An, Yue-Ie Hsing, Moo Young Eun, Chang-Deok Han, Narayana Upadhyaya, Srinivasan Ramachandran, Qifa Zhang, Andy Pereira, Venkatesan Sundaresan, and Hei Leung. Rice mutant resources for gene discovery. *Plant Mol Biol*, 54(3) :325–334, Feb 2004.

[HK04] Thomas Hernandez and Subbarao Kambhampati. Integration of biological sources : Current systems and challenges ahead. *SIGMOD Record*, 33(3) :51–60, 2004.

[Hor99] Ian Horrocks. FaCT and iFaCT. In *Description Logics*, 1999.

[Hor02] Ian Horrocks. DAML+OIL : a description logic for the semantic web. *IEEE Data Engineering Bulletin*, 25(1) :4–9, 2002.

[HSK+01] L. M. Haas, P. M. Schwarz, P. Kodali, E. Kotlar, J. E. Rice, and W. C. Swope. Discoverylink : a system for integrated access to life sciences data sources. *IBM Syst. J.*, 40(2) :489–511, 2001.

[HSO94] K. W. Hart, David B. Searls, and G. Christian Overton. Sortez : a relational translator for ncbi's asn.1 database. *Computer Applications in the Biosciences*, 10(4) :369–378, 1994.

[HWS+06] Duncan Hull, Katy Wolstencroft, Robert Stevens, Carole Goble, Mathew R Pocock, Peter Li, and Tom Oinn. Taverna : a tool for building and running workflows of services. *Nucleic Acids Res*, 34(Web Server issue) :W729–W732, Jul 2006.

[IKJ+07] Katica Ilic, Elizabeth A Kellogg, Pankaj Jaiswal, Felipe Zapata, Peter F Stevens, Leszek P Vincent, Shulamit Avraham, Leonore Reiser, Anuradha Pujar, Martin M Sachs, Noah T Whitman, Susan R McCouch, Mary L Schaeffer, Doreen H Ware, Lincoln D Stein, and Seung Y Rhee. The plant structure ontology, a unified vocabulary of anatomy and morphology of a flowering plant. *Plant Physiol*, 143(2) :587–599, Feb 2007.

[Ini00] Arabidopsis Genome Initiative. Analysis of the genome sequence of the flowering plant arabidopsis thaliana. *Nature*, 408(6814) :796–815, Dec 2000.

[Inm96] W.H. Inmon. *Building the Data Warehouse*. John Wiley & sons, inc., New York, 1996.

[Int05a] International Rice Genome Sequencing Project. The map-based sequence of the rice genome. *Nature*, 436 :793–800, 2005.

[Int05b] International Rice Research Institute. Wild rice taxonomy. http : // www.knowledgebank.irri.org / wildricetaxonomy, 2005.

[JAI+05] Pankaj Jaiswal, Shulamit Avraham, Katica Ilic, et al. Plant ontology (po) : a controlled vocabulary of plant structures and growth stages. *Comparative and Functional Genomics*, 6(7-8) :388–397, 2005. doi :10.1002/cfg.496.

[JHG+05] Alexander A T Johnson, Julian M Hibberd, Céline Gay, Pauline A Essah, Jim Haseloff, Mark Tester, and Emmanuel Guiderdoni. Spatial control of transgene expression in rice (oryza sativa l.) using the gal4 enhancer trapping system. *Plant J*, 41(5) :779–789, Mar 2005.

[JJW+05] Yu J, Wang J, Lin W, Li S, and Li H. The genomes of oryza sativa : a history of duplications. *PLoS Biol*, 3(2) :e38, 2005.

[JM61] F. JACOB and J. MONOD. Genetic regulatory mechanisms in the synthesis of proteins. *J Mol Biol*, 3 :318–356, Jun 1961.

[JPS+06] Andrew R Jones, Angel Pizarro, Paul Spellman, Michael Miller, and FuG. E. Working Group. Fuge : Functional genomics experiment object model. *OMICS*, 10(2) :179–184, 2006.

[KAA+07] Tamara Kulikova, Ruth Akhtar, Philippe Aldebert, Nicola Althorpe, Mikael Andersson, and et al. Embl nucleotide sequence database in 2006. *Nucleic Acids Res*, 35(Database issue) :D16–D20, Jan 2007.

[Kar95] P.D. Karp. A strategy for database interoperation. *J Comput Biol*, 2(4) :573–86, 1995.

[KCVGC+05] Ingrid M Keseler, Julio Collado-Vides, Socorro Gama-Castro, John Ingraham, Suzanne Paley, Ian T Paulsen, Martín Peralta-Gil, and Peter D Karp. Ecocyc : a comprehensive database resource for escherichia coli. *Nucleic Acids Res*, 33(Database issue) :D334–D337, Jan 2005.

[KFG84] M. Kanehisa, J. W. Fickett, and W. B. Goad. A relational database system for the maintenance and verification of the los alamos sequence library. *Nucleic Acids Res*, 12(1 Pt 1) :149–158, Jan 1984.

[KFNM04] H. Knublauch, R. W. Fergerson, N. F. Noy, and M. A. Musen. The protégé owl plugin : An open development environment for semantic web applications. In *Third International Semantic Web Conference, Hiroshima, Japan*, 2004.

[KG00] M. Kanehisa and S. Goto. Kegg : kyoto encyclopedia of genes and genomes. *Nucleic Acids Res*, 28(1) :27–30, Jan 2000.

[KKS+04] Arek Kasprzyk, Damian Keefe, Damian Smedley, Darin London, William Spooner, Craig Melsopp, Martin Hammond, Philippe Rocca-Serra, Tony Cox, and Ewan Birney. Ensmart : a generic system for fast and flexible access to biological data. *Genome Res*, 14(1) :160–169, Jan 2004.

[KOMK+05] Peter D Karp, Christos A Ouzounis, Caroline Moore-Kochlacs, Leon Goldovsky, Pallavi Kaipa, Dag Ahrén, Sophia Tsoka, Nikos Darzentas, Victor Kunin, and Núria López-Bigas. Expansion of the biocyc collection of pathway/genome databases to 160 genomes. *Nucleic Acids Res*, 33(19) :6083–6089, 2005.

[KP96] P. D. Karp and S. Paley. Integrated access to metabolic and genomic data. *J Comput Biol*, 3(1) :191–212, 1996.

[KPKZ04] P. D. Karp, S. Paley, C. J. Krieger, and P. Zhang. An evidence ontology for use in pathway/genome databases. *Pac Symp Biocomput*, pages 190–201, 2004.

[KRPPT02] Peter D Karp, Monica Riley, Suzanne M Paley, and Alida Pellegrini-Toole. The metacyc database. *Nucleic Acids Res*, 30(1) :59–61, Jan 2002.

[KRS+00] P. D. Karp, M. Riley, M. Saier, I. T. Paulsen, S. M. Paley, and A. Pellegrini-Toole. The ecocyc and metacyc databases. *Nucleic Acids Res*, 28(1) :56–59, Jan 2000.

[KZM+04] Cynthia J Krieger, Peifen Zhang, Lukas A Mueller, Alfred Wang, Suzanne Paley, Martha Arnaud, John Pick, Seung Y Rhee, and Peter D Karp. Metacyc : a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Res*, 32(Database issue) :D438–D442, Jan 2004.

[LBW+04] P. Lord, S. Bechhofer, M. Wilkinson, G. Schiltz, D. Gessler, D. Hull, C. Goble, and L. Stein. Applying semantic web services to bioinformatics : Experiences gained, 2004.

[LDB+04] Rasko Leinonen, Federico Garcia Diez, David Binns, Wolfgang Fleischmann, Rodrigo Lopez, and Rolf Apweiler. Uniprot archive. *Bioinformatics*, 20(17) :3236–3237, Nov 2004.

[Lev99] Alon Y. Levy. Combibining artificial intelligence and databases for data integration. In *Artificial Intelligence Today*, pages 249–268, 1999.

[LGL+07] Pierre Larmande, Céline Gay, Mathias Lorieux, Christophe Périn, Matthieu Bouniol, Gaëtan Droc, Christophe Sallaud, Pascual Perez, Isabelle Barnola, Corinne Biderre-Petit, Jérôme Martin, Jean Benoît Morel, Alexander A T Johnson, Fabienne Bourgis, Alain Ghesquière, Manuel Ruiz, Brigitte Courtois, and Emmanuel Guiderdoni. Oryza tag line, a phenotypic mutant database for the genoplante rice insertion line library. *Nucleic Acids Res*, Oct 2007.

[LMNR04] Zoé Lacroix, Hyma Murthy, Felix Naumann, and Louiqa Raschid. Links and paths through life sciences data sources. In *DILS*, pages 203–211, 2004.

[LRV04] Zoé Lacroix, Louiqa Raschid, and Maria-Esther Vidal. Efficient techniques to explore and rank paths in life science data sources. In *DILS*, pages 187–202, 2004.

[MAA+05] Nicola J Mulder, Rolf Apweiler, Teresa K Attwood, Amos Bairoch, Alex Bateman, David Binns, and et al. Interpro, progress and status in 2005. *Nucleic Acids Res*, 33(Database issue) :D201–D205, Jan 2005.

[MBFS02] Ioana Manolescu, Luc Bouganim, Françoise Fabret, and Eric Simon. Efficient querying of distributed resources in mediator systems. pages 468–485, 2002.

[MBFS03] Ioana Manolescu, Luc Bouganim, Françoise Fabret, and Eric Simon. Interrogation efficace de ressources distribuées dans des systèmes de médiation. *Technique et Science Informatiques*, 22(10) :1271–1296, 2003.

[McK89] V. A. McKusick. Hugo news. the human genome organisation : history, purposes, and membership. *Genomics*, 5(2) :385–387, Aug 1989.

[MCKS97] V. M. Markowitz, I. M. Chen, A. S. Kosky, and E. Szeto. Facilities for exploring molecular biology databases on the web : a comparative study. *Pac Symp Biocomput*, pages 256–267, 1997.

[MDC03] E. Motta, J. Domingue, and L. Cabral. Irs-ii : A framework and infrastructure for semantic web services, 2003.

[MEC07] Christopher J Mungall, David B Emmert, and FlyBase Consortium. A chado case study : an ontology-based modular schema for representing genome-associated biological information. *Bioinformatics*, 23(13) :i337–i346, Jul 2007.

[MFS+86] K. Mullis, F. Faloona, S. Scharf, R. Saiki, G. Horn, and H. Erlich. Specific enzymatic amplification of dna in vitro : the polymerase chain reaction. *Cold Spring Harb Symp Quant Biol*, 51 Pt 1 :263–273, 1986.

[MG77] A. M. Maxam and W. Gilbert. A new method for sequencing dna. *Proc Natl Acad Sci U S A*, 74(2) :560–564, Feb 1977.

[MGB04] J.-F. Morot-Gaudry and J.-F. Briat. *La génomique en biologie végétale*. INRA, 2004.

[MGF05] Josette Masle, Scott R Gilmore, and Graham D Farquhar. The erecta gene regulates plant transpiration efficiency in arabidopsis. *Nature*, 436(7052) :866–870, Aug 2005.

[MHTH01] P. Mork, A. Halevy, and P. Tarczy-Hornoch. A model for data integration systems of biomedical data applied to online genetic databases. *Proc AMIA Symp*, pages 473–477, 2001.

[MIK+07] Akio Miyao, Yukimoto Iwasaki, Hidemi Kitano, Jun-Ichi Itoh, Masahiko Maekawa, Kazumasa Murata, Osamu Yatou, Yasuo Nagato, and Hirohiko Hirochika. A large-scale collection of phenotypic data describing an insertional mutant population to facilitate functional analysis of rice genes. *Plant Mol Biol*, 63(5) :625–635, Mar 2007.

[Min75] M. Minsky. A framework for representing knowledge. *In The Psychology of Computer Vision, edited by P.H. Winston (New York : McGraw-Hill)*, pages 211–277, 1975.

[MJMN62] J. H. MATTHAEI, O. W. JONES, R. G. MARTIN, and M. W. NIRENBERG. Characteristics and composition of rna coding units. *Proc Natl Acad Sci U S A*, 48 :666–677, Apr 1962.

[MKL+05] Malika Mahoui, Harshad Kulkarni, Nianhua Li, Zina Ben-Miled, and Katy Börner. Semantic correspondence in federated life science data integration systems. In *DILS*, pages 137–144, 2005.

[MKZ+88] SR McCouch, G Kochert, Yu ZH, ZY Wang, GS Khush, WR Coffman, and SD Tanksley. Molecular mapping of rice chromosomes. *Theor Appl Genet*, 76 :815–829, 1988.

[MMJN62] R. G. MARTIN, J. H. MATTHAEI, O. W. JONES, and M. W. NIRENBERG. Ribonucleotide composition of the genetic code. *Biochem Biophys Res Commun*, 6 :410–414, Jan 1962.

[Moo95] G. Moore. Cereal genome evolution : pastoral pursuits with 'lego' genomes. *Curr Opin Genet Dev*, 5(6) :717–724, Dec 1995.

[Mor10] T. H. Morgan. Sex-limited inheritance in *Drosophila*. *Science*, 32 :120–122, 1910.

[MSMB15] T. H. Morgan, A.H. Sturtevand, H.J. Muller, and C. Bridges. The mechanism of mendelian heredity. *Henry Holt and Co., New York*, 1915.

[MSTH05] P. Mork, R. Shaker, and P. Tarczy-Hornoch. The multiple roles of ontologies in the biomediator data integration system. In *DILS*, 2005.

[NFM00] N. F. Noy, R. W. Fergerson, and M. A. Musen. The knowledge model of protege-2000 : Combining interoperability and flexibility. In *2th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000) Juan-les-Pins, France,*, 2000.

[NL05] Pieter B T Neerincx and Jack A M Leunissen. Evolution of web services in bioinformatics. *Brief Bioinform*, 6(2) :178–188, Jun 2005.

[NM00] Natalya Fridman Noy and Mark A. Musen. PROMPT : Algorithm and tool for automated ontology merging and alignment, 2000.

[NSD+01] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protege-2000. In *IEEE Intelligent Systems*, volume 16(2), pages 60–71, 2001.

[OAF+04] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R Pocock, Anil Wipat, and Peter Li. Taverna : a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17) :3045–3054, Nov 2004.

[OAS] OASIS. Web services flow language (wsfl) (url :http ://xml.coverpages.org/wsfl.html).

[OMG07] OMG. Otology definition metamodel, 09 2007. OMG Document Number : ptc/2007-09-09.

[OWL03] OWL-S Coalition. Owl-s 1.0 release. (url : http//www.daml.org/services/owl-s/1.0/), 2003.

[PCF+06] Joshua Phillips, Ram Chilukuri, Gilberto Fragoso, Denise Warzel, and Peter A Covitz. The cacore software development kit : streamlining construction of interoperable biomedical information services. *BMC Med Inform Decis Mak*, 6 :2, 2006.

[PJK+06] Anuradha Pujar, Pankaj Jaiswal, Elizabeth A Kellogg, Katica Ilic, Leszek Vincent, Shulamit Avraham, Peter Stevens, Felipe Zapata, Leonore Reiser, Seung Y Rhee, Martin M Sachs, Mary Schaeffer, Lincoln Stein, Doreen Ware, and Susan McCouch. Whole-plant growth stage ontology for angiosperms and its application in plant biology. *Plant Physiol*, 142(2) :414–428, Oct 2006.

[PMFR92] P. L. Pearson, N. W. Matheson, D. C. Flescher, and R. J. Robbins. The gdb human genome data base anno 1992. *Nucleic Acids Res*, 20 Suppl :2201–2206, May 1992.

[PS96] Christine Parent and Stefano Spaccapietra. Intégration de bases de données : Panorama des problèmes et des approches. *Ingéniérie des systèmes d'information*, 4(3) :1–18, 1996.

[RAH+96] Mary Tork Roth, Manish Arya, Laura M. Haas, Michael J. Carey, William F. Cody, Ronald Fagin, Peter M. Schwarz, Joachim Thomas II, and Edward L. Wimmers. The garlic project. In *SIGMOD Conference*, page 557, 1996.

[RBB+07] Paolo Romano, Ezio Bartocci, Guglielmo Bertolini, Flavio De Paoli, Domenico Marra, Giancarlo Mauri, Emanuela Merelli, and Luciano Milanesi. Biowep : a workflow enactment portal for bioinformatics applications. *BMC Bioinformatics*, 8 Suppl 1 :S19, 2007.

[RBG+97] A. L. Rector, S. Bechhofer, C. A. Goble, I. Horrocks, W. A. Nowlan, and W. D. Solomon. The grail concept modelling language for medical terminology. *Artif Intell Med*, 9(2) :139–171, Feb 1997.

[RKO+03] Anthony Rowe, Dimitrios Kalaitzopoulos, Michelle Osmond, Moustafa Ghanem, and Yike Guo. The discovery net system for high throughput bioinformatics. *Bioinformatics*, 19 Suppl 1 :i225–i231, 2003.

[RN94] A. L. Rector and W. A. Nowlan. The galen project. *Comput Methods Programs Biomed*, 45(1-2) :75–78, Oct 1994.

[RRZ+03] A. L. Rector, J. E. Rogers, P. E. Zanstra, E. Van Der Haring, and OpenG. A. L. E. N. Opengalen : open source medical terminology and tools. *AMIA Annu Symp Proc*, page 982, 2003.

[SAUT+02] A. Sasaki, M. Ashikari, M. Ueguchi-Tanaka, H. Itoh, A. Nishimura, D. Swapan, K. Ishiyama, T. Saito, M. Kobayashi, G. S. Khush, H. Kitano, and M. Matsuoka. Green revolution : a mutant gibberellin-synthesis gene in rice. *Nature*, 416(6882) :701–702, Apr 2002.

[SBB+00] R. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N. W. Paton, C. A. Goble, and A. Brass. Tambis : transparent access to multiple bioinformatics information sources. *Bioinformatics*, 16(2) :184–185, Feb 2000.

[SBB+02] Jason E Stajich, David Block, Kris Boulez, Steven E Brenner, Stephen A Chervitz, Chris Dagdigian, Georg Fuellen, James G R Gilbert, Ian Korf, Hilmar Lapp, Heikki Lehväslaiho, Chad Matsalla, Chris J Mungall, Brian I Osborne, Matthew R Pocock, Peter Schattner, Martin Senger, Lincoln D Stein, Elia Stupka, Mark D Wilkinson, and Ewan Birney. The bioperl toolkit : Perl modules for the life sciences. *Genome Res*, 12(10) :1611–1618, Oct 2002.

[SGB00] R. Stevens, C. A. Goble, and S. Bechhofer. Ontology-based knowledge representation for bioinformatics. *Brief Bioinform*, 1(4) :398–414, Nov 2000.

[SGHB02] Robert Stevens, Carole Goble, Ian Horrocks, and Sean Bechhofer. Building a bioinformatics ontology using oil. *IEEE Trans Inf Technol Biomed*, 6(2) :135–141, Jun 2002.

[SGL+04] Christophe Sallaud, Céline Gay, Pierre Larmande, Martine Bès, Pietro Piffanelli, Benoit Piégu, Gaétan Droc, Farid Regad, Emmanuelle Bourgeois, Donaldo Meynard, Christophe Périn, Xavier Sabau, Alain Ghesquière, Jean Christophe Glaszmann, Michel Delseny, and Emmanuel Guiderdoni. High throughput t-dna insertion mutagenesis in rice : a first step towards in silico reverse genetics. *Plant J*, 39(3) :450–464, Aug 2004.

[SHX+05] Sohrab P Shah, Yong Huang, Tao Xu, Macaire M S Yuen, John Ling, and B. F Francis Ouellette. Atlas - a data warehouse for integrative bioinformatics. *BMC Bioinformatics*, 6 :34, 2005.

[Sim01] Eric Simon. Le select, a middleware system that eases the publication of scientific data sets and programs. In *Workshop on Information Integration on the Web*, page 2, 2001.

[SLR+03] C. Sallaud, M. Lorieux, E. Roumen, D. Tharreau, R. Berruyer, P. Svestasrani, O. Garsmeur, A. Ghesquiere, and J-L. Notteghem. Identification of five new blast resistance genes in the highly blast-resistant rice variety ir64 using a qtl mapping strategy. *Theor Appl Genet*, 106(5) :794–803, Mar 2003.

[SMS+02] Paul T Spellman, Michael Miller, Jason Stewart, Charles Troup, Ugis Sarkans, Steve Chervitz, Derek Bernhart, Gavin Sherlock, Catherine Ball, Marc Lepage, Marcin Swiatek, W. L. Marks, Jason Goncalves, Scott Markel, Daniel Iordan, Mohammadreza Shojatalab, Angel Pizarro, Joe White, Robert Hubley, Eric Deutsch, Martin Senger, Bruce J Aronow, Alan Robinson, Doug Bassett, Christian J Stoeckert, and Alvis Brazma. Design and implementation of microarray gene expression markup language (mage-ml). *Genome Biol*, 3(9) :RESEARCH0046, Aug 2002.

[SMvB+03] C. Sallaud, D. Meynard, J. van Boxtel, C. Gay, M. Bès, J. P. Brizard, P. Larmande, D. Ortega, M. Raynal, M. Portefaix, P. B F Ouwerkerk, S. Rueb, M. Delseny, and E. Guiderdoni. Highly efficient production and characterization of t-dna plants for rice (oryza sativa l.) functional genomics. *Theor Appl Genet*, 106(8) :1396–1408, May 2003.

[SNC77] F. Sanger, S. Nicklen, and A. R. Coulson. Dna sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A*, 74(12) :5463–5467, Dec 1977.

[SOA] Recommandation du w3c sur soap (simple object access protocol) (url :http ://www.w3.org/tr/soap12-testcollection/).

[SRG03] Robert D Stevens, Alan J Robinson, and Carole A Goble. myGrid : personalised bioinformatics on the information grid. *Bioinformatics*, 19 Suppl 1 :i302–i304, 2003.

[Sta05] World Rice Statistics. Irri, los baños, philippines, http :// www.irri.org/ science/ ricestat/, 2005.

[Ste03] Lincoln D Stein. Integrating biological databases. *Nat Rev Genet*, 4(5) :337–345, May 2003.

[Suj01] W. Sujansky. Heterogeneous database integration in biomedicine. *J Biomed Inform*, 34(4) :285–298, Aug 2001.

[SWC+95] W. Y. Song, G. L. Wang, L. L. Chen, H. S. Kim, L. Y. Pi, T. Holsten, J. Gardner, B. Wang, W. X. Zhai, L. H. Zhu, C. Fauquet, and P. Ronald. A receptor kinase-like protein encoded by the rice disease resistance gene, xa21. *Science*, 270(5243) :1804–1806, Dec 1995.

[TAI04] R. Terada, H. Asao, and S. Iida. A large-scale agrobacterium-mediated transformation procedure with a strong positive-negative selection for gene targeting in rice (oryza sativa l.). *Plant Cell Rep*, 22(9) :653–659, Apr 2004.

[TRM+05] Silke Trissl, Kristian Rother, Heiko Müller, Thomas Steinke, Ina Koch, Robert Preissner, Cornelius Frömmel, and Ulf Leser. Columba : an integrated database of proteins, structures, and annotations. *BMC Bioinformatics*, 6 :81, 2005.

[TRV98] Anthony Tomasic, Louiqa Raschid, and Patrick Valduriez. Scaling access to heterogeneous data sources with DISCO. *Knowledge and Data Engineering*, 10(5) :808–823, 1998.

[VAM+01] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, and et al. The sequence of the human genome. *Science*, 291(5507) :1304–1351, Feb 2001.

[VBV99] Eric Viara, Emmanuel Barillot, and Guy Vaysseix. The eyedb oodbms. In *IDEAS*, pages 390–402, 1999.

[W3Ca] W3C. Web services / services web (definition) (url : http ://www.w3.org/2002/ws/).

[W3Cb] W3C. web services activities (url : http ://www.w3.org/2002/ws/).

[W3Cc] W3C. Web services choregraphy description language version 1.0 (url : http ://www.w3.org/tr/ws-cdl-10/).

[W3Cd] W3C. Web services choregraphy interface (wsci) (url :http ://www.w3.org/tr/wsci/).

[W3Ce] W3C. Wsdl (web services description language) (url : http ://www.w3c.org/tr/wsdl/).

[WAM+03] Denise B Warzel, Christo Andonaydis, Bill McCurry, Ram Chilukuri, Sadritdin Ishmukhamedov, and Peter Covitz. Common data element (CDE) management and deployment in clinical trials. *AMIA Annu Symp Proc*, page 1048, 2003.

[WC53] J. D. WATSON and F. H. CRICK. Molecular structure of nucleic acids ; a structure for deoxyribose nucleic acid. *Nature*, 171(4356) :737–738, Apr 1953.

[Wie92] Gio Wiederhold. Mediators in the arhiecture of future information systems. In *IEEE Computer*, pages 38–49, March 1992.

[Wil06] Mark Wilkinson. Gbrowse moby : a web-based browser for biomoby services. *Source Code Biol Med*, 1 :4, 2006.

[WJN+02] Doreen H Ware, Pankaj Jaiswal, Junjian Ni, Immanuel V Yap, Xioakang Pan, Ken Y Clark, Leonid Teytelman, Steven C Schmidt, Wei Zhao, Kuan Chang, Sam Cartinhour, Lincoln D Stein, and Susan R McCouch. Gramene, a tool for grass genomics. *Plant Physiol*, 130(4) :1606–1613, Dec 2002.

[WL02] Mark D Wilkinson and Matthew Links. BioMOBY : an open source biological web services proposal. *Brief Bioinform*, 3(4) :331–341, Dec 2002.

[WMB+94] G. L. Wang, D. J. Mackill, J. M. Bonman, S. R. McCouch, M. C. Champoux, and R. J. Nelson. Rflp mapping of genes conferring complete and partial resistance to blast in a durably resistant rice cultivar. *Genetics*, 136(4) :1421–1434, Apr 1994.

[WSEH05] Mark Wilkinson, Heiko Schoof, Rebecca Ernst, and Dirk Haase. BioMOBY successfully integrates distributed heterogeneous bioinformatics Web Services. The PlaNet exemplar case. *Plant Physiol*, 138(1) :5–17, May 2005.

[WSGA03] C. J. Wroe, R. Stevens, C. A. Goble, and M. Ashburner. A methodology to migrate the gene ontology to a description logic environment using daml+oil. *Pac Symp Biocomput*, pages 624–635, 2003.

[WSM] WSMF. Web service modeliing framework (wsmf) (url :http ://www1-c703.uibk.ac.at/users/c70385/wese/index.html).

[xli05] *XLive : An XML Light Integration Virtual Engine*, Saint-Malo, France, 2005.

[Xu02] Y Xu. *Global view of QTL : rice as a model. In : Quantitative genetics, genomics and plant breeding*. CAB International, 2002.

[YHW+02] Jun Yu, Songnian Hu, Jun Wang, Gane Ka-Shu Wong, Songgang Li, Bin Liu, and et al. A draft sequence of the rice genome (oryza sativa l. ssp. indica). *Science*, 296(5565) :79–92, Apr 2002.

[YJ05] Yukiko Yamazaki and Pankaj Jaiswal. Biological ontologies in rice databases. an introduction to the activities in gramene and oryzabase. *Plant Cell Physiol*, 46(1) :63–68, Jan 2005.

[YKA+00] M. Yano, Y. Katayose, M. Ashikari, U. Yamanouchi, L. Monna, T. Fuse, T. Baba, K. Yamamoto, Y. Umehara, Y. Nagamura, and T. Sasaki. Hd1, a major photoperiod sensitivity quantitative trait locus in rice, is closely related to the arabidopsis flowering time gene constans. *Plant Cell*, 12(12) :2473–2484, Dec 2000.

[YKNA03] Iwei Yeh, Peter D Karp, Natalya F Noy, and Russ B Altman. Knowledge acquisition, consistency checking and concurrency control for gene ontology (go). *Bioinformatics*, 19(2) :241–248, Jan 2003.

[ZFT+05] Peifen Zhang, Hartmut Foerster, Christophe P Tissier, Lukas Mueller, Suzanne Paley, Peter D Karp, and Seung Y Rhee. Metacyc and aracyc. metabolic pathway databases for plant research. *Plant Physiol*, 138(1) :27–37, May 2005.

[ZLAE02] Evgeni M Zdobnov, Rodrigo Lopez, Rolf Apweiler, and Thure Etzold. The ebi srs server–recent developments. *Bioinformatics*, 18(2) :368–373, Feb 2002.

[ZLW+06] Jianwei Zhang, Caishun Li, Changyin Wu, Lizhong Xiong, Guoxing Chen, Qifa Zhang, and Shiping Wang. Rmd : a rice mutant database for functional analysis of the rice genome. *Nucleic Acids Res*, 34(Database issue) :D745–D748, Jan 2006.

Table des figures

Résumé : Les recherches que nous présentons dans ce mémoire s'inscrivent dans le cadre des problématiques d'intégration de données hétérogènes en génomique fonctionnelle végétale. La génomique fonctionnelle se définit en biologie comme un cadre dans lequel plusieurs disciplines et techniques participent à la découverte de la fonction des gènes. Elle génère un volume important de données que les scientifiques gèrent de manières diverses. Or de nombreuses sources de données, qu'elles soient complémentaires ou chevauchantes, sont nécessaires pour enrichir les informations sur la fonction des gènes. Le problème est qu'elles sont stockées de manière distribuées, autonomes avec plusieurs niveaux d'hétérogénéité, laissant le biologiste chercher l'information et intégrer les résultats manuellement.

L'objectif de cette thèse est de permettre aux scientifiques d'accéder de manière transparente aux informations issues de plusieurs sources de données de génomique fonctionnelle. Pour cela nous nous proposons d'aborder deux approches afin d'en évaluer les avantages et les inconvénients. Premièrement nous proposons l'intégration de sources à travers l'adaptation d'un système de médiation de données et de programmes : Le Select. Successeur de DISCO, Le Select permet l'intégration de sources de données hétérogènes et distribuées avec un modèle pivot relationnel. Deuxièmement, nous proposons la création d'un environnement utilisateur personnalisé intégrant les sources à travers des enchaînements de services Web. Ce système est basé sur l'application BioMOBY et son annuaire de services bioinformatiques. Pour conclure, fort de l'expérience menée sur l'intégration et le partage, nous proposons une méthodologie adaptée aux besoins d'intégration pour des projets analogues.

Mots-clés : Génomique fonctionnelle végétale, intégration, médiation, services Web, BioMOBY, LE SELECT, bioinformatique.

Abstract : In this document, we present research topic developed in the context of heterogeneous data integration in plant functional genomic. Plant functional genomic is a biological framework where several disciplines and techniques take part in the discover of genes function. It generates a large quantity of data which the scientists manage in various ways. However, many data sources, complementary or overlapping, are necessary to enrich information about genes function. The problem comes from the distribution, the autonomy and the heterogeneity of these sources. That drags biologists seeking information, to integrate results manually.

The objective of this thesis is to make easier the scientists searches and to reach in a transparent way information resulting from several data sources. For that, we propose two approaches in order to evaluate the advantages and the disadvantages of them. Firstly we propose the integration of sources through the adaptation of a mediation system : Select. Successor of DISCO, Le Select allows the integration of heterogeneous and distributed data sources through a relationnal integration model. Secondly, we propose the creation of a user personalized environment that integrate data sources through workflows of Web services. This system is based on BioMOBY system and its Central Registry. To conclude, we propose a methodology adapted to the needs for similar integration projects.

Keywords : Plant functional genomic, data integration, middleware, Web services, BioMOBY, LE SELECT, bioinformatics.